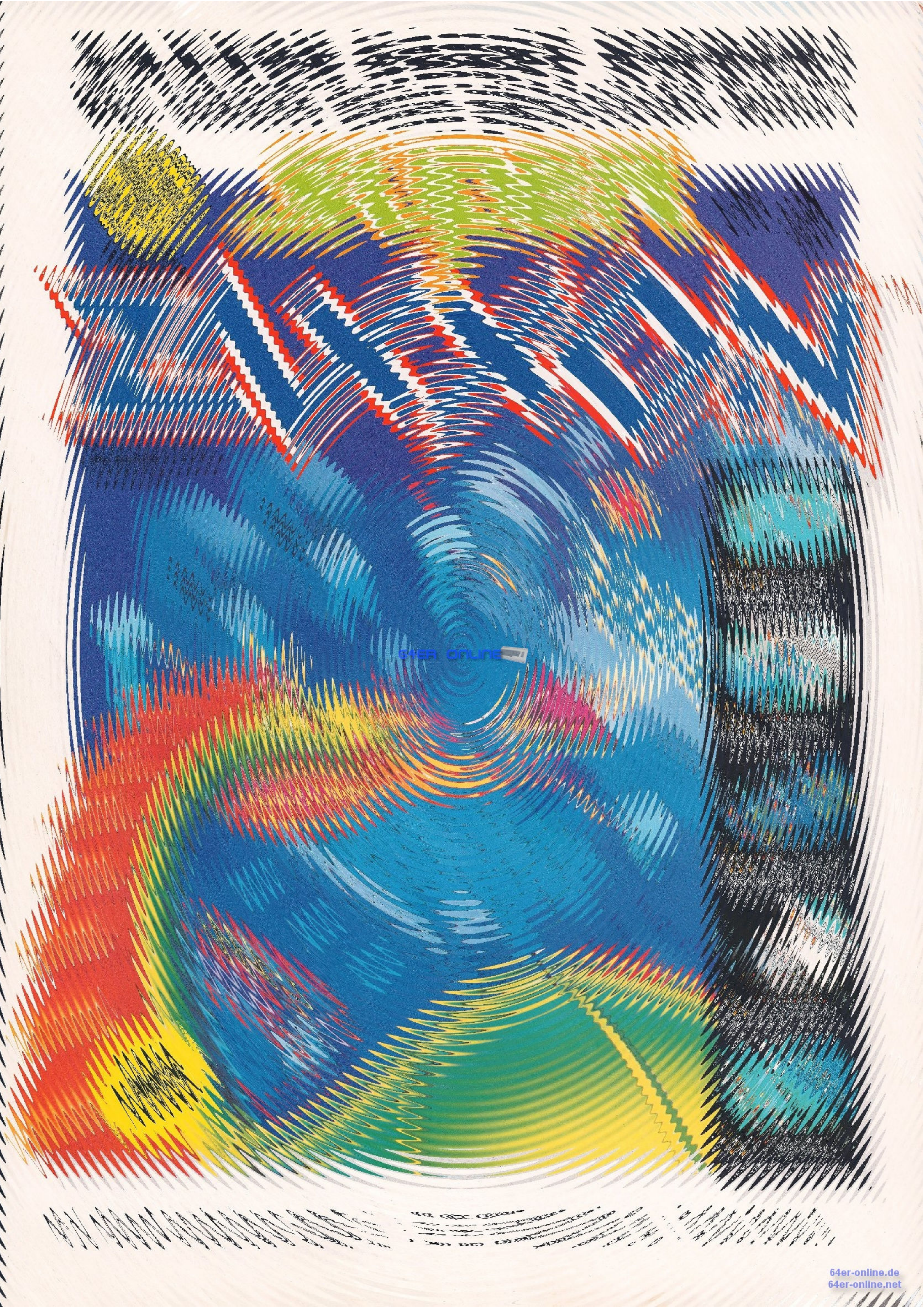


Markt&Technik
JANUAR 1986
ÖS 55,– /Sfr. 6,50
Lit 5.500/hfl 8,– /dkr 30,–
DM 6,50
64'er
1/86
DAS MAGAZIN FÜR COMPUTER-FANS
Ausführliche Testberichte
★ C 128 D
★ Floppy 1571
★ Profi-Software für C 128
Spiele: Alles über Simulationen
Der richtige Monitor
★ Tips für den Kauf
★ Selbstbau: besserer Ton, besseres Bild
Zum Abtippen: Superschneller Tiny-Basic-Compiler
128D
Tips und Tricks für C 64 und C 128
Neuer Kurs: Von Basic zu Assembler
★ 64'er Extra: alle Kernel-Routinen
★ DFÜ-News ★ So programmiert man Hardcopies
64er-online.de
64er-online.net





Aktuell

Hardware-Test

Der richtige Monitor

Spiele

128er-Sonderteil

Wettbewerbe

Listing zum Abtippen

Seite 40



Viermal Schachmatt

Das Turnier ist beendet, die Sieger und Verlierer stehen fest. Vier Schachprogramme sind gegeneinander angetreten, um ihre Spielstärke zu zeigen. Und um das Turnier komplett zu machen, ließen wir die Programme noch gegen einen Meister seiner Klasse spielen: den Schachcomputer »Mephisto«.
Seite 40

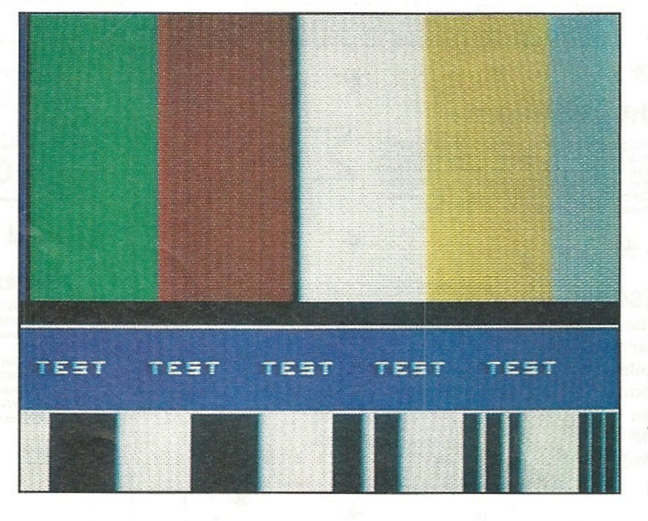
Seite 52



Datawork-Basic

Schaffen Sie Ordnung in Ihrem Notizzettel-Berg! Mit Datawork-Basic, unserer neuen Anwendung des Monats, programmieren Sie spielend leicht die effektvollsten Dateiverwaltungen. 22 neue Befehle zur problemlosen Maskenerstellung, für Diskettenzugriffe und Definition von Datenfeldern helfen Ihnen dabei.
Seite 52

Seite 24



Großer Monitor-Test

Wir haben für Sie 15 Monitore unter die Lupe genommen. In einem ausführlichen Bericht erfahren Sie die Testergebnisse der Monochrom- und Color-Typen. Welche Monitore passen zum C 128 und welche zum C 64? Bei jedem Monitor werden die positiven und negativen Eigenschaften herausgestellt. Lassen Sie sich überraschen!
Seite 24

Seite 44

Die neuen Floppy-Laufwerke

C 128-Besitzer können aufatmen: Die neuen Laufwerke sind endlich lieferbar, man muß sich nicht mehr mit der langsamen 1541 herumschlagen. Aber welches der beiden neuen Geräte, die 1570 oder die 1571, soll man sich zulegen? Unser Test hilft Ihnen weiter.

Seite 44



Seite 37

Simulationen

Kein anderes Software-Genre verursacht soviele Kontroversen wie die Heimcomputer-Simulationen. Haben sie nur Unterhaltungswert oder sind sie ernsthaft nutzbar? Wie stehen sie im Vergleich zu industriellen Groß-Simulationen da? Sind sie wirklich nur »Spiele«?

Seite 37



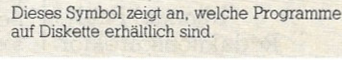
Seite 20

Drucker vom Uhrenhersteller

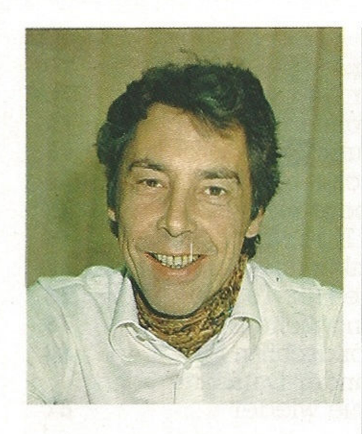
Mit der MSP-Reihe wollen die Japaner den Druckermarkt erobern. Unser MSP-10-Test zeigt wie die Chancen für Citizen stehen. Mit dem Riteman II stellen wir Ihnen einen Drucker der neuen Kompaktklasse vor. Sie werden erstaunt sein, welche Leistungen sich in einem kleinen Gehäuse verbergen.

Seite 20





▪ Dieses Symbol zeigt an, welche Programme auf Diskette erhältlich sind.



Tausch-Tips

Einzelne Softwareanbieter verfolgen ihre Rechte rigoros. Ein beliebter Anhaltspunkt für juristische Aktionen sind die Tauschlisten, die viele Benutzer zusammengestellt haben und an Interessenten verschicken. Sie werden daraufhin ausgewertet, ob möglicherweise Raubkopien angeboten werden. Mangels Kenntnissen und Erfahrungen bei Polizei, Rechts- und Staatsanwälten kommen dabei oft auch Computerbenutzer in arge Schwierigkeiten, denen gar nichts vorzuwerfen ist. Unnötigen Problemen beim Softwaretausch kann leicht aus dem Weg gehen, wer bei dem Zusammenstellen von Tauschlisten außer dem Programmnamen jeweils auch den Autor angibt — und ggf. auch noch, ob mit dem Betreffenden eine Weitergabe vereinbart ist, oder ob es sich um sogenannte »Freeware« oder Public Domain Software handelt. Wer auch seine Disketten entsprechend beschriftet hat (und auf Back-up-Disketten vermerkt: Sicherungskopie), hat zwei Vorteile: Wenn wirklich einmal die Polizei auf der Suche nach Raubkopien auftauchen sollte, kann er leicht demonstrieren, daß hier alles seine Ordnung hat. Er kann aber damit rechnen, daß Tauschlisten mit den oben empfohlenen Angaben ohnehin niemand veranlassen werden, hier nachzufassen. Wer nur eigene Programme zum Tausch anbietet oder solche, die nicht urheberrechtlich geschützt sind, sollte mit seiner Tauschliste ruhig demonstrieren, daß er ein reines Gewissen hat.

Michael Pauly,
Redaktionsdirektor

Hobby Elektronik-Messe zum Anfassen

Das Schwabenland war schon immer dafür bekannt, das Land der Bastler und Tüftler zu sein. So kam es denn auch, daß man vom 7. bis 10. November auf dem Stuttgarter Killesberg eine Messe erleben konnte, die sich nicht nun in ihrer Zielsetzung von anderen Messen wie beispielsweise der Systems unterschiedet. Zunächst fiel auf, daß man so gut wie alles Ausgestellte auch gleich einpacken und mit nach Hause nehmen konnte. Wichtigstes Thema waren und sind auf dieser Messe aber Computer und die Menschen, die mit ihnen umgehen. Zu den Ausstellern gehörten deshalb hauptsächlich kleine und größere Händler, Hardware-Entwicklungshäuser (Dela, Kalawsky), einige Fachzeitschriften sowie ein paar wenige Giganten des Handels wie Horten, Schreiber Elektronik, Mirwald und Vobis.

Neben den interessanten Diskussionen direkt am Computer (es wurde selbst auf der Messe noch eifrig programmiert) konnte man auch so manches »Schnäppchen« machen. Da war beispielsweise ein winziger Stand eines Bremer Händlers, der ständig umringt war, mit gutem Grund: Es gab leicht angeschlagene Computer vom VC 20 für 25 Mark bis zum Osborne Executive für 1900 Mark. Oder der Abverkauf der C16/C 116-Computer, die man am Vobisstand einiges unter 200 Mark bekommen konnte. Gleiches gilt auch für den Atari 800 XL, der mit 238 Mark einen Preis erreichte, bei dem jedem, der noch vor einem Jahr fast 1000 Mark dafür bezahlt hatte, die Tränen in die Augen hochstiegen. Gleich gegenüber gab es sogar eine Europapremiere am Stand von Weber Computertechnik. Dort zeigte man den brandneuen Citizen 120 D (noch neuer als die gerade erst auf der Systems vorgestellte MSP-Reihe). Dieser handliche Drucker scheint das Zeug zu haben, einer der ganz großen zu werden, denn er bietet beinahe alle Leistungen der fast doppelt so teuren MSP-Reihe (NLQ-Schrift, Hexdump mit ASCII-Wert, Grafikfähigkeit, 4 KByte Pufferspeicher). Dazu soll wahlweise ein Centronics- oder Commodore-Einschubmodul mit Schnittstelle erhältlich sein. Das kleine Wunderding soll nach Angaben des Ausstellers für unter 1000 Mark erhältlich sein, entsprechend groß war der Andrang vor dem Ausstellungsstück (Test in Ausgabe 2/86). Erstaunlich auch die Tatsache, daß dieser neue Drucker auf der Systems noch nicht zu sehen war, obwohl der Generalimporteur für Citizen, nämlich Synelec, dort einen großen Ausstellungsstand hatte. Da stellt sich natürlich die Frage, ob der 120 D nun die Nachhut der MSP-Reihe ist, oder etwa ein Vorbote einer neuen Druckergeneration mit hervorragenden Leistungen zu niedrigsten Preisen?

Mirwald Electronic aus Unterhaching setzte neben der altbekannten Produktpalette (BMC, Epson) wieder auf die neue Picom-Druckgeneration. LQ 80 (137 Zeichen pro Zeile) und LQ 130 (232 Zeichen pro Zeile) heissen sie, stellen sich mit einer Schönschrift (18 x 20 Punktmatrix) und einer Geschwindigkeit von bis zu 180 Zeichen pro Sekunde vor. Dabei erinnern die Picom-Drucker nicht nur äußerlich an einen FX-85 Drucker, sondern weisen auch einen ähnlichen Befehlssatz auf. Der Preis liegt bei 1598 Mark (LQ 80) beziehungsweise 1998 Mark (LQ 130).

Maximaler Miniplotter

Bei Hack Elektronik gab es neben einem winzigen (Zigarettenschachtelformat) Fernseher auch Interessantes aus der Plotter-Szene zu sehen. Der MCP-80 ist ein A4-Flachbettplotter mit vier Farben und einer Centronics-Schnittstelle (seriell RS232 auf Wunsch). Alle Farben sollen sich sowohl durch Tastendruck als auch per Software ansteuern lassen. Der Preis von 798 Mark scheint für einen Plotter mit einer Wiederholgenauigkeit von 0,2 mm relativ günstig.

MS-DOS wo man hinschaut

Unverkennbar war der Trend zu ernsthaften Computeranwendungen wie Datenverwaltung und Textverarbeitung an fast allen Ständen. Spitzenreiter der in diesem Bereich angebotenen Computer sind die sogenannten »IBM-Kompatiblen«. An fast allen Ständen konnte man diese Computer in den verschiedensten Ausbaustufen bewundern. Bemerkenswert ist dabei das Preisniveau. Einen einfachen MS-DOS-Computer mit einem Laufwerk konnte man schon für 1888 Mark mit nach Hause nehmen. Selbst in voller Ausbaustufe, sogar mit einem 10 MByte Festplattenlaufwerk, mußte nur wenig mehr als 5000 Mark bezahlt werden. Inwieweit nun alle IBM-Programme auf diesen meist in Fernost produzierten Computern auch lauffähig sind, konnten wir natürlich nicht austesten. Wer auf garantierte Funktionsfähigkeit aller MS-DOS-Programme Wert legt, konnte am Schreiber Elektronic-Stand eine echte Preissensation gleich mitnehmen — einen tragbaren Original IBM-PC mit zwei Laufwerken, eingebautem Monitor und 256 KByte RAM, dessen Preis von zirka 7000 Mark auf 3990 Mark gesenkt wurde. Klar, daß es sich dabei um ein Auslaufmodell handelt, aber sicher nicht um eine schlechte Investition.

Ach ja, gespielt wurde auch auf der Hobby Elektronik und zwar fast ausschließlich auf einem, mit Computern exzellent ausgestatteten, MSX-Stand. So wie es schien, liegt der Schwerpunkt dieses japanischen Standardisierungsversuchs darin, eine Spielkonsole mit Computer-Alibifunktion darzustellen. Das allerdings machen die MSX-Computer, wie man sehen konnte, gar nicht schlecht.

Nicht vergessen werden soll aber die Vielzahl der kleinen und kleinsten Ausstellungsstände, auf denen man sich so richtig ins Detail hineinfragen konnte. So geschehen auf dem Stand von Kalawsky, der nicht nur seinen ausgezeichneten EPROM-Brenner vorstellte, sondern auch eine Vielzahl kleinerer und größerer Erweiterungen für den C 64 und den C 128. Dazu gehören eine akkugepufferte Echtzeituhr, ein am User-Port betriebenes Digital-Volt-Meter, ein Testgerät für EPROM-Programmiergeräte und ein User-Port-Verdoppler.

Rundum gelungen

Natürlich war die Hobby Elektronik keine Ausstellung, bei der es darauf ankam, im Scheinwerferlicht die neueste Produktpalette vorgestellt zu bekommen. Ganz im Gegenteil, hier trafen sich die Profis und solche, die es werden wollen. Wer mit etwas Ruhe durch die Hallen gegangen ist, konnte so manchen guten Tip aufschnappen und was als nachahmenswert einmalig gelten kann, sofort ausprobieren. Als Messe »zum Anfassen« hat sich ein Besuch auf der Hobby Elektronik in jedem Fall gelohnt. Nirgendwo war die Demonstration von dem, was man mit den Worten »Computererlebnis live« beschreiben könnte, größer als in Stuttgart. (aw)

Neue Softwae von Softline: Jet und Newsroom

Die Firma Softline bietet zwei neue Software-Produkte für den C 64 an. Das eine ist das Programm »Newsroom«, mit dem man Zeitungen erstellen kann. Texte werden mit einem Texteditor geschrieben, Bilder mit einem Grafikeditor erstellt. Beides wird dann zu einer richtigen Zeitung verbunden. Ausgegeben wird das Ganze auf einem Matrixdrucker. Verschiedene Zeichensätze und über 600 fertige Grafiken unterstützen die »Zeitungsmacher«.

Ebenfalls neu ist »Jet«, der neue Flugsimulator von SubLogic, der Firma, die durch den »Flight Simulator II« weltberühmt wurde. »Jet« simuliert einen modernen Kampfjet.

Eine Meldung am Rande: Die ersten Programme für den Amiga werden demnächst bei Softline lieferbar sein. Fest angekündigt sind »Crimson Crown« und »Transylvania«, zwei besonders gute Grafik-Adventures. Softline will bei Amiga-Software weiterhin am Ball bleiben, so daß bei der Markteinführung im Frühjahr 1986 bereits viele Programme zur Verfügung stehen sollen. (bs)

Info: Softline, Schwarzwaldstr. 8a, 7602 Oberkirch

Zwei neue Spiele von Lucasfilm

Bei Activision sind demnächst zwei neue Spiele der Firma Lucasfilms erhältlich. Bei »Koronis Rift« muß der Spieler auf einem Planeten nach Überresten einer versunkenen Zivilisation suchen. »The Eidolon« versetzt den Spieler ins vorherige Jahrhundert und von dort in eine fantastische Welt der Drachen und Trolle. Beide Spiele verwenden die von »Rescue on Fractalus« bekannte 3D-Grafik. Sie sind sehr aktionsreich, bieten aber auch viele logische und strategische Elemente und fantastisch ausgearbeitete Hintergrundgeschichten. In dem ausklingenden Weihnachtsgeschäft könnte sie sich zu Rennern entwickelt. (bs)

Info: Activision Deutschland GmbH, Karlstr. 26, 2000 Hamburg 76

Elite-Wettbewerb abgebrochen

Wer sich in den letzten Wochen ein deutschsprachiges Elite-Original gekauft hat, konnte darin eine Karte für eine deutsche Elite-Meisterschaft finden. Damit sollten innerhalb von sechs Monaten die besten Elite-Spieler Deutschlands ermittelt werden. Diese Meisterschaft mußte nun komplett abgesagt werden. Der Grund: In den ausgelieferten Exemplaren befindet sich ein Fehler, der es möglich macht, beliebig viel Geld zu erreichen und seinen Rang zu verbessern, indem man ein paar Tasten drückt und sich somit viele Stunden spannenden Spiels »erspart«. Da also ein »Betrug« bei dem Wettbewerb möglich und nicht nachweisbar wäre, mußte leider komplett und ersatzlos gestrichen werden. (bs)

Video-Filmschnitt mit dem C 64

Creative Video bietet ein Zusatzgerät zum C 64 an, das zwei Videorecorder koppelt. Die C 64-Video-Schnittsteuerung soll ein einfaches und extrem genaues Schneiden von Videofilmen ermöglichen. Sie wird an den User-Port des C 64 angeschlossen. Am Videorecorder sind keine Eingriffe nötig.

Der Schnittvorgang wird von der mitgelieferten Software übernommen.

Die Schnittgenauigkeit ist nach Angaben des Herstellers bei VHS- Rekordern besser als 300 ms Filmdauer. Als Aufnahmerecorder ist jeder Videorecorder mit Start/Stop-Anschluß und Assemble-Schnitt geeignet. Als Wiedergaberecorder sind besonders geeignet: JVC HR-C3, Telefunken 900M, Saba CVR 6073, ITT VRP 3844, Graetz TR 4844 und Dual.

Creative Video bietet ein Interface einschließlich Software auf Diskette zum Preis von 494 Mark an. (cg)

Info: Creative Video, Postfach 1501, 8520 Erlangen, Telefon (091 95)-2728

Computer, Peripherie, Meßtechnik

Der neue »PK elektronic« Katalog 86/87 bietet auf 72 Seiten Abbildungen und technische Daten zu Geräten der Hersteller Fluke, Metrix, Toellner, Epson, Facit, Qume, Tele Video und Selanar. Das Angebot von »PK elektronic« richtet sich vor allem an professionelle Anwender in Labor, Service und Forschung.

Der Katalog soll kostenlos bei PK elektronik erhältlich sein. (cg)

Info: PK elektronic Poppe GmbH, Lietzenburger Str. 91, 1000 Berlin 15, Telefon (030) 883 1058

Zwei Computer an einen Drucker

Für 248 Mark gibt es bei Wiesemann einen Adapter, um zwei Computer an einen einzigen Drucker mit Centronics-Schnittstelle anzuschließen.

Der neue Schnittstellen-Vervielfacher 2C/C/0 schaltet automatisch zwischen zwei Computern (Centronics-Schnittstellen) um. Der Computer, der zuerst anfängt auszudrucken, blockiert sofort den anderen. Erst wenn der gerade druckende Drucker einige Sekunden keine Daten mehr sendet, kann der andere anfangen zu drucken. Eine unerwünschte Mischung der Texte soll ausgeschlossen sein. (cg)

Info: Wiesemann Mikrocomputertechnik, Winchenbachstraße 3-5, 5600 Wuppertal 2, Telefon (0202) 505077

Schnelles DOS ohne Kabelsalat

Inzwischen wird ein Floppyspeeder, der ohne Parallelübertragungskabel funktioniert, angeboten. Nicht nur Programm-Transfers werden beschleunigt, sondern auch Zugriffe auf sequentielles Files. Lediglich das Betriebssystem-ROM und ein Floppy-ROM müssen ausgewechselt werden. Laden geht dann sechs- bis siebenmal schneller als bisher, speichern etwa zwei bis dreimal schneller. Der Zugriff auf sequentielle Files und Direktzugriff mit Block-Read- und Block-Write-Befehlen geht nun zwei- bis dreimal schneller als bisher.

Das Programm bietet außerdem ein Floppy-DOS ähnlich dem DOS 5.1. Die F-Tasten sind belegt mit Directory-Laden, LOAD, RUN und LIST.

Drückt man während eines Reset die SPACE-Taste, entkommt man jedem Programm.

Mit dem Floppy-Befehl »UI-« kann man auf normale Ladegeschwindigkeit zurückschalten.

Die RS232-Routinen sind noch im Betriebssystem vorhanden und weiterhin verwendbar.

Geliefert werden zwei EPROMs mit Adaptersockel und Kurzanleitung. Der Preis für das schnelle DOS liegt bei 79 Mark. (Manfred Kohlen/cg)

Info: Bernd Stevermüer, Kurfürstenstraße 62, 6050 Offenbach

Computer-TV-Adapter für die Video-Buchse des Farbfernsehers

ViTec bietet ein Videokabel für alle Computer mit 5poliger Buchse an. Bei Computern mit Cinch-Buchse ist ein zusätzlicher Stecker-Übergang nötig. Der Adapter verbindet den Heimcomputer mit der Videobuchse des TV-Geräts. Mit einem Schalter kann man zwischen Fernsehempfang und Computer umschalten.

Der Computer-Adapter ist ausgelegt für TV-Geräte mit der 6poligen DIN-AV-Buchse. Für TV-Geräte mit der 21poligen SCART-Buchse gibt es einen zusätzlichen Adapter.

Der Computer-Adapter 4 C 6 kostet 38 Mark. Der Stecker-Übergang Nr. 4 (für Cinch-Buchsen) ist für 14 Mark zu haben. Der SCART-Adapter (Nr. 9) kann für 29 Mark bezogen werden. (cg)

Info: ViTec Audio-Video GmbH, Postfach 57, 3003 Ronnenberg 1, Telefon (051 09)-1400

Commodore für Prominente

Heinz Schenk, bekannt durch seine Sendung »Blauer Bock«, interessiert sich augenscheinlich nicht nur für Äppelwoi. Auf der Commodore Fachaustellung in Frankfurt hat sich Heinz Schenk ausführlich über das Angebot an Soft- und Hardware rund um den C 64/C 128 informiert. Begleitet wurde er dabei vom Commodore-Geschäftsführer Winfried Hoffmann und Pressesprecher Gerold Hahn.

Herr Schenk arbeitet aktiv mit dem C 64 und ist mittlerweile auf den C 128 umgestiegen. (aa)

Winfried Hoffmann, Heinz Schenk und Gerold Hahn (von links nach rechts) beim Studium der besten Zeitschriften, die es für Heimcomputer gibt. ▼



Systems '85

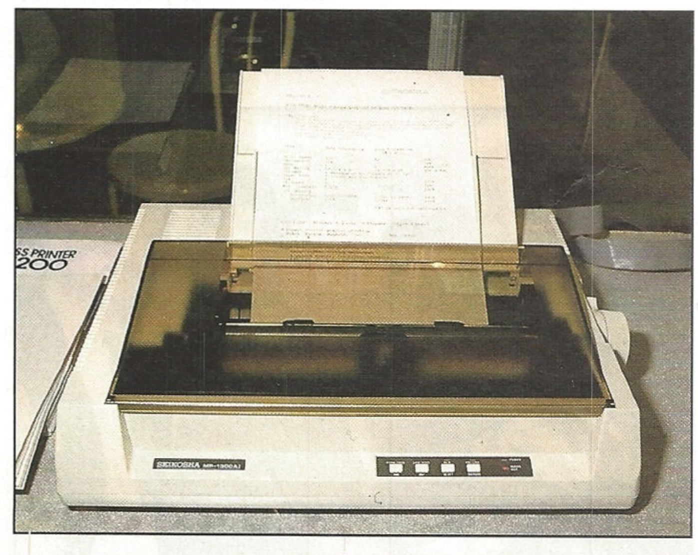
Auf der Systems in München war einiges Interessantes zu sehen. Viele Hersteller scheinen die Farbe entdeckt zu haben. So wurden dann auch eine ganze Palette farbtüchtiger Drucker vorgestellt. Dabei dominierten zwei wesentliche Konstruktionsprinzipien: Einmal das bekannte Prinzip der reinen Farbdrucker, die entweder auf Thermo-Transfer-Basis oder mit einem mehrfarbigen Nylon-Farbband arbeiten. Zum anderen aber auch Drucker einer neuen, flexiblen Generation. Es handelt sich dabei um Hochleistungsdrucker, die normalerweise mit einem schwarzen Farbband arbeiten und alle Anforderungen an einen Alltagsdrucker (NLQ-Schrift, bis zu 300 Zeichen pro Sekunde) erfüllen, bei Bedarf aber mit wenigen Handgriffen zum Farbdrucker umgebaut werden können.

Bei Fujitsu konnte man einen solchen Hochleistungsdrucker, den DX 2100, ausprobieren. Er soll in der Normalschrift bis zu 220 Zeichen pro Sekunde drucken können und auch in der NLQ-Schrift noch eine ausgezeichnete Geschwindigkeit erreichen. Der DX 2100 ist ein 9-Nadel-Matrixdrucker mit Centronics-Schnittstelle, der innerhalb kürzester Zeit vom Anwender in einen Farbdrucker umgebaut werden kann. Durch seine Befehlskompatibilität zum Epson FX-80 stehen einer Anwendung zusammen mit vielen Anwenderprogrammen kaum Hindernisse im Weg. Trotz dieser Leistungsfähigkeit ist der DX 2100, der übrigens einen sehr soliden Eindruck macht, relativ preiswert. Der empfohlene Verkaufspreis wird ohne Umrüstsatz für Farbe bei zirka 1950 Mark liegen. Der Preis für den Farb-Umrüstsatz stand noch nicht fest.



64'er Redakteur A. Wängler erklärt die Vorteile des C 128

Bei Seikosha hat man sich etwas besonderes einfallen lassen. Im Rahmen einer japanischen Kulisse wurde der MP 1300 AI vorgestellt. Es stellte sich bald heraus, daß der Aufwand wohl begründet war, denn der MP 1300 AI ist ein wahres Rennpferd. In der Normalschrift soll er mit bis zu 300 Zeichen pro Sekunde schreiben und sogar in der ebenfalls vorhandenen NLQ-Schrift bleiben davon noch immerhin 60 Zeichen pro Sekunde übrig. Diese Vorteile bleiben auch erhalten, wenn man den MP 1300 AI, ebenfalls mit wenigen Handgriffen, zum Farbdrucker umbaut. Danach stehen immerhin sieben Farben zur Verfügung, die aus einem vierfarbigen Farbband gewonnen werden. Auch dieser Drucker wird mit Centronics- oder wahlweise serieller RS232-Schnittstelle geliefert und kann mit einem Interface an den C 64 angeschlossen werden. Ein weiteres außergewöhnliches Merkmal des MP 1300 AI ist sein Pufferspeicher von 10 KByte, der sogar noch erweitert werden kann.



Der Schnelle: Bis zu 300 Zeichen pro Sekunde schafft der MP 1300 AT

Der Preis wird zum Erscheinungstermin im Frühjahr 1986 nach Aussage des Herstellers unter 2000 Mark liegen.

Den dritten im Bunde dieser neuen Drucker-Philosophie hatte Juki ausgestellt. Der 5510, ein Matrixdrucker, soll Geschwindigkeit von 180 Zeichen/Sekunde haben und zum Farbdrucker umgerüstet werden können. Unter der Bezeichnung 5520 wird derselbe Drucker mit eingebautem Farb-Set angeboten. Auch bei diesem Drucker ist die, in dieser Preisklasse mittlerweile unverzichtbare NLQ-Schrift vorhanden. Die Geschwindigkeit soll bei dieser Schriftart immerhin noch 30 Zeichen pro Sekunde betragen.

Für etwa 1500 Mark (Modell 5510) beziehungsweise 1700 Mark (Modell 5520) ist dieser 8,5 Kilogramm schwere Drucker bereits im Fachhandel erhältlich.

Obwohl schon der kürzlich vorgestellte Epson FX-85 durch sein sehr gutes Schriftbild glänzte, so hat er im vollkommen neuen LQ-800 seinen Meister gefunden. Dieser kleine Bruder des bereits seit über einem Jahr erhältlichen Super-Druckers LQ 1500 hat ein Schriftbild, bei dem man das »Near« vor der »Letter Quality« fast weglassen kann. Der LQ-800 ist mit seinen 24 Nadeln ein wahrer Schönschreiber. Dabei erreicht er mit 60 Zeichen pro Sekunde in dieser Schrift immer noch ein vielfaches der meisten Typenraddrucker. Das Schriftbild ist so perfekt, daß sogar die normalerweise sehr heikle Kursivschrift ohne sichtbare Abstufungen gedruckt wird. Auf der Ausstellung wurde gezeigt, daß vom bewährten Prinzip der Druckersteuerung durch ESC-Befehle auch beim LQ-800 nicht abgewichen wurde. Der Anschluß des LQ-800 an den C 64/C 128 der mit einer Centronics-Schnittstelle ausgerüstet ist, ist mit den gleichen Interface-Modulen wie beim FX-85 möglich. Ein ganz besonderes Leistungsmerkmal des LQ-800 ist sein neues Moduleinschubsystem. Dadurch sollen die Variations- und Ausbaufähigkeiten des Druckers enorm erweitert werden. Diese Module enthalten weitere standardisierte Schriftarten, die sich mit den bekannten Befehlen aber ihrerseits wieder mannigfaltig beeinflussen lassen. Ähnlich flexibel sollen die Grafikfähigkeiten des LQ-800 sein. Mit einer Auflösung von bis zu 360 Punkten pro Zoll wird eine wesentlich bessere Darstellung von Grafiken als beim FX-85 mit seinen 240 Punkten pro Zoll möglich. Der LQ-800 wird nach Angaben von Epson einen empfohlenen Verkaufspreis von unter 2500 Mark haben.

Wer behauptet, daß alle Matrixdrucker mehr oder weniger laut sein müssen, wurde am Stand von Mannesmann vom Gegenteil überzeugt. Der MT 85 ist mit unter 52 dB (A) leiser als der normale Bürolärm oder der heimische Fernsehapparat. Damit sind aber die Leistungsmerkmale des MT 85 noch nicht erschöpft. Ähnlich wie beim LQ-800 besitzt auch der Mannesmann einen Modulschacht, in dem er die verschiedensten Schriftarten aufnehmen kann. Auch hier stehen danach wieder alle gängigen Manipulationen der Schriftart zur Verfügung. Besonderer Reiz an allem ist, daß unabhängig vom gewählten Modul, jede Schrift auch in einer NLQ-Version zur Verfügung steht. Mit seinem umfangreichen Befehlssatz haben sich die deutschen Entwickler im übrigen an die Befehle des Epson FX-85 gehalten, dessen Grafikfähigkeiten ebenfalls vorhanden sind. Ausgestattet mit einer Centronics-Schnittstelle liegt der Preis des 180 Zeichen pro Sekunde schnelle MT 85 ungefähr bei 1600 Mark. (aw)

Videobilder farbig digitalisieren

Füle Electronic Trading stellte einen Video-Digitizer für den C 64 vor. Es soll das erste Seriengerät sein, das farbig digitalisieren kann. Die damit erzielten Resultate sind verblüffend. Die Bilder werden im Koalapainter-Format im C 64 angelegt, so daß sie mit diesem Zeichenprogramm nachbearbeitet werden können. Der Digitizer wird an den User-Port angeschlossen. Die Steuersoftware wird mitgeliefert. Der Preis: Mit etwa 400 Mark wird er nur unwesentlich teurer als ein Schwarzweiß-Videodigitizer sein. (bs)

Info: Füle Electronic Trading, Birkenstr. 22, 6057 Dietzenbach 1

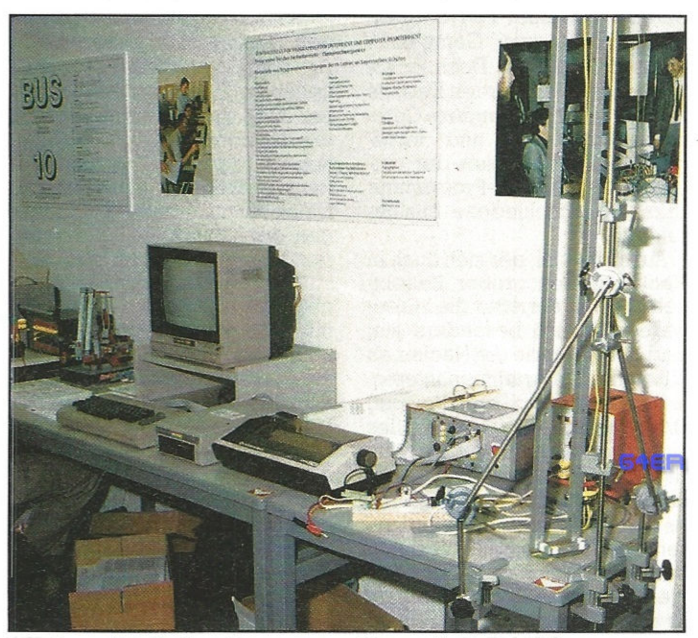
Computer für den Unterricht

Diese Meldung betrifft alle Lehrer und Schulen, die für ihren Unterricht Computer einsetzen wollen. Die Zentralstelle für Computer im Unterricht vermittelt Programme aus den unterschiedlichsten Unterrichtsbereichen. Auf der Systems wurden einige interessante Physikexperimente mit dem C 64 vorgeführt. Das Bild zeigt einen Aufbau zur Ermittlung der Erdbeschleunigung.

Von der Zentralstelle kann man die dazu benötigten Gerätschaften zwar nicht beziehen, aber man bekommt Informationen darüber, wo was bezogen werden kann und wie teuer es ist. In der Zentralstelle für Computer im Unterricht werden auch Programme gesammelt, die dann an Schulen weitergegeben werden. Alle Programme werden mit einer ausführlichen Dokumentation versehen, wobei man versucht einen einheitlichen Standard in der Beschreibung zu erreichen. Nähere Informationen erteilt die Zentralstelle für Computer im Unterricht.

Scherlingstr. 7-9
8900 Augsburg
Telefon: 0821/573011

(hm)



Die Zentralstelle für Computer im Unterricht vermittelt Unterrichtshilfen

Fernseh-Tuner für Ihren Monitor

Wollen Sie Ihren Monitor in einen Fernseher verwandeln? Mit dem TV-Tuner »High Screen« von Vobis ist das kein Problem. Der High-Screen ist ein TV-Empfangsteil, das die Signale der Sendeanstalten monitorgerecht aufbereitet. Zwölf VHF- und UHF-Programme können eingestellt und auf Tastendruck abgerufen werden. Die eingebaute AFC-Schaltung übernimmt die Feinabstimmung.

Für einen guten Empfang kann der »High Screen« an eine Dachantenne oder einen Antennenverteiler angeschlossen werden. Falls kein Antennenanschluß verfügbar ist, hilft die mitgelieferte Stabantenne weiter. Sie wird von hinten auf das Tuner-Gehäuse aufgesteckt.

Der »High Screen« hat neben dem Antenneneingang auch einen Video- (FBAS) und Audio-Eingang, an dem ein Computer oder ein Videorecorder angeschlossen werden kann (Cinch-Buchsen). Mit einem Schalter an der Frontseite kann man zwischen TV-Empfang und Computer oder Videorecorder umschalten.

Als Monitor läßt sich jeder Farbmonitor mit FBAS- und Audio-Eingang verwenden. Der Anschluß erfolgt über ein etwa 1 Meter langes Kabel mit Cinch-Steckern.

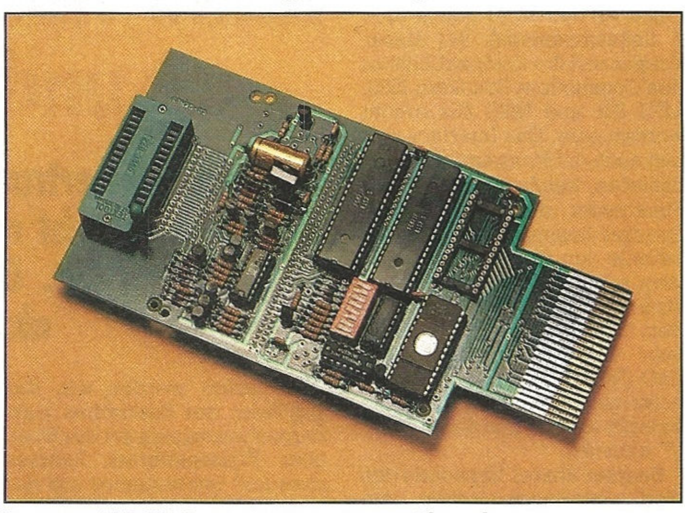
Der Tuner läßt sich auch für »reinrassige« Videorecorder verwenden, die keinen Tuner eingebaut haben, wenn man statt des Monitors einen Recorder anschließt.

Wir testeten den »High Screen« mit einem 1702-Monitor. Der Bildempfang und die Bildqualität waren überraschend gut.

(hm)

Info: Vobis, Viktoriastr. 74, 5100 Aachen, Tel. 0241/500081
Einführungspreis: 249 Mark

Neues EPROM-Programmiergerät von Thienel



Das neue EPROM-Programmiergerät von Thienel

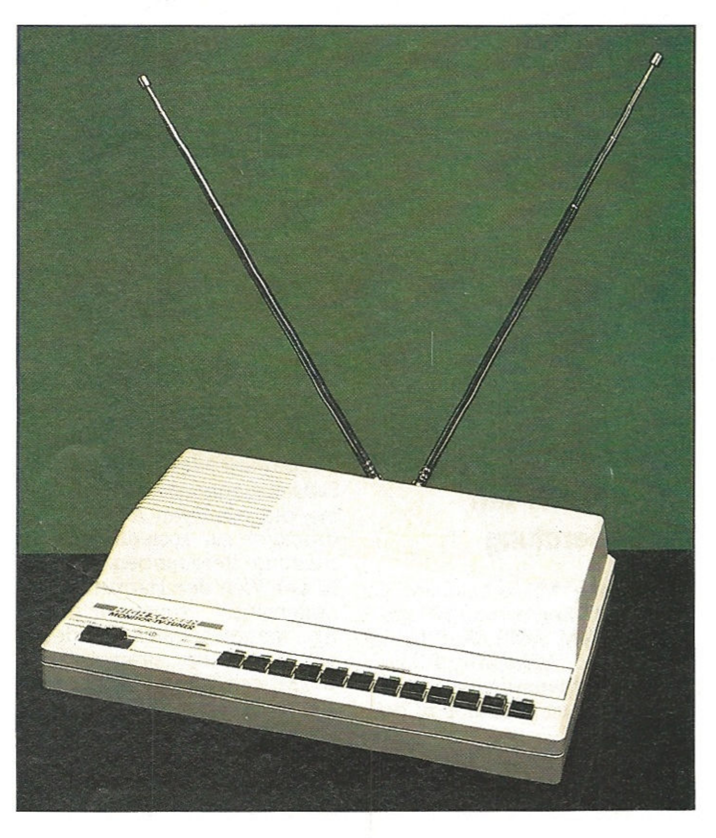
Von Stefan Thienel gibt es einen neuen PEROM-Brenner, der den in Ausgabe 7/85 getesteten ablöst (Bild).

Programmiert werden können die EPROM-Typen: 2516 bis 2564, 2716 bis 27512L/H. Beim Programmieren kann zwischen 50 Millisekunden und einem variablen Algorithmus gewählt werden.

Das Steuerprogramm befindet sich in einem EPROM auf der Platine des EPROM-Programmiergeräts, das am Expansions-Port angeschlossen wird (vergoldete Kontakte). Als EPROM hierfür kann ein 2764 oder 27128 verwendet werden. Über DIP-Schalter wird der EPROM-Typ eingestellt. Beim »27128« kann man wählen, ob man die gesamten 16 KByte laden will, oder nur den unteren oder oberen 8-KByte-Teil. Ebenso bei dem zusätzlichen freien Stecksockel auf der Platine, in dem man beispielsweise gebrannte EPROMs auf Funktionsfähigkeit testen kann. Zusätzlich zum Steuerprogramm ist im EPROM ein Modulgenerator und ein Monitorprogramm untergebracht.

Das EPROM-Programmiergerät benötigt eine Betriebsspannung von etwa 30 bis 35 Volt Wechselstrom.

Info: Stefan Thienel, Sandgasse 22, Tel. 09546/1456
Bezugsquelle: Steinmetz-Elektronik, Nürnberger Str. 49, 8600 Bamberg, 0951/202638.
Preis: Bausatz 169 Mark, Fertiggerät 199 Mark.



Die High-Screen-Video-Tuner von Vobis

Neues Görlitz-Interface

Die Version 3.7 (3. Hardware-, 7. Softwareversion). Bei dieser Version soll die Kompatibilität zu den Commodore-Druckern 1525, MPS 801 und MPS 802 erhöht worden sein. Das Interface soll nun auch mit Programmen funktionieren, welche die Tabulator-Funktionen der Commodore-Drucker ansprechen. Folgende Befehle von Commodore-Druckern sollen jetzt auch vom Görlitz-Interface ausgewertet werden: Tabulator: CHR$(16), Tabulator in Einzelpunktschritten: ESC CHR$(16) und das Wiederholkommando: CHR$(26). Diese Befehle werden erst durch ESC" V" aktiviert.

Besitzer älterer Versionen des Görlitz-Interfaces sollen ein EPROM mit der neuesten Software zum Preis von 20 Mark bekommen. Im Austausch gegen das alte EPROM. Laut Wiesemann reicht es, 20 Mark (als Verrechnungsscheck) zusammen mit dem alten EPROM einzuschicken. Die neue Version sollen Sie dann postwendend zugesandt bekommen. (hm)

Info: Görlitz Computerbau, Postfach 852, 5400 Koblenz, 0261/2044

Im Gesellschaftsspiel Basic lernen

Aus Ungarn kommt ein Kartenspiel, mit dem man wirklich spielend Basic lernen kann. Dazu wird kein Computer benötigt. »AssBasic« kann in drei Schwierigkeitsgraden gespielt werden. Im ersten kann wirklich jeder mitspielen, der noch nie etwas von Basic gehört hat, auch kleine Kinder. In einer Art »Schwarzer Peter« lernt man dann die »Bedeutung« der Befehle RUN, PRINT etc. kennen. In den beiden weiteren Spielstufen wird das Wissen vertieft und Programme geschrieben. Wohlbemerkt, alles ohne Computer! Die Umsetzung ins Deutsche wird gerade vorbereitet, deswegen wird »AssBasic« erst in einigen Wochen erhältlich sein. (bs)

Info: Füle Electronic Trading, Birkenstr. 22, 6057 Dietzenbach 1

Katalog für EDV-Zubehör mit Telefonberatung

Der neue Misco-Katalog '85 bietet auf 56 Seiten rund 800 Produkte von 3 M, AMP, BASF, DEC, Dysan, IBM, Memorex, Rhone Poulenc, Verbatim und Misco an. Der Katalog ist kostenlos zu beziehen. (cg)

Info: Misco GmbH, Nordendstraße 72-74, 6082 Mörfelden-Walldorf, Telefon (06105) 4010

Hallo Hacker, Mailbox-Freaks und DFÜ-Freunde!

Datenfernübertragung wird immer mehr zum Volkssport. Dazu braucht man aber solide Informationen und so manchen Tip. Die B.H.P. und die 64'er haben für Sie aktuelles zusammengestellt.

Ab sofort werden wir, das Team der Bayrischen Hakkerpost zusammen mit der 64'er, über Wissenswertes, Lehrreiches und Interessantes aus der Welt der Netzwerke und Mailboxen, der Host-Computer und Terminals berichten.

Unter anderem ist vorgesehen:

- Einführung in Datex-P-Benutzung
- Berichte über Akustikkoppler und sonstiges DFÜ-Zubehör
- Bücher rund ums Hacken
- DFÜ-Programme im Test
- Aktuelles aus der Szene
- Kummerkasten für Anfänger und Fortgeschrittene.

Uns erreicht man entweder per DFÜ über Datex-P in folgenden Boxen:

- COM.BOX Berlin (NUA 45300040509 unter 64-Redaktion)
- DECATES Oberramstadt (NUA 45615440307 unter BHP)
- RMI Aachen (NUA 44241040341 unter BHP)
- ZEV Zürich (NUA 022847911118 unter BHP).

Über Telefon im Umkreis München:
Commodore Anwender Club München (Tel. 089/8120338).
Und per Bildschirmtext, Leitseite *92049204#.

Zu guter Letzt auch noch über den normalen Postweg:

B.H.P.
c/o Basis
Adalbertstr. 41b
8000 München 40.

Kopieren erlaubt — Freeware!

Freeware oder Public Domain Software nennen sich Programme, die von ihren Autoren ausdrücklich zur kostenfreien Verbreitung freigegeben wurden. In der Welt der Hacker ist das bekannteste Programm dieser Art wahrscheinlich Modem7, ein Terminalprogramm unter CP/M, das von Ward Christensen geschrieben und von zahllosen Benutzern weiterentwickelt wurde.

Ein weiteres, sehr weit verbreitetes Terminalprogramm ist Kermit. Kermit wurde von der Columbia University in New York entwickelt und diente zunächst nur zur Übertragung von Dateien von einer PDP-11 zu einer DEC-20, zwei Computern aus der Groß-EDV. Doch innerhalb kurzer Zeit wurden Kermits auch für allerhand andere Computer entwickelt, und inzwischen umfaßt die Liste der vorhandenen Kermit-Programme zirka 100 verschiedene Ausführungen.

Auf dem C 64, der sich auch in Hackerkreisen großer Beliebtheit erfreut, sprießte die »Freeware« natürlich besonders gut, sind doch gerade die Hacker an nützlichen Programmen interessiert. So gibt es zum Beispiel »Megacom«, ein universelles Hackprogramm, das selbständig Nummern anruft und protokolliert oder Töne erzeugt, mit denen man umsonst telefonieren kann. Leider geht das nur in den USA, da das Telefonnetz hierzulande anders aufgebaut ist als in Amerika. Wer seine eigene Mailbox mit C 64 und 1541 aufmachen will, der ist mit »Norad« bestens bedient. Für das andere Ende gibt es »Connection 64«, »Term V1.7«, »Videx« oder »Proterm 64«.

Und was die Herzen aller Hakker höher schlagen läßt: »Data Hacker«, ein sehr komfortables und leistungsfähiges Programm zum automatischen Durchprobieren von Datex-P-Nummern und Paßwörtern.

Alle obengenannten Programme (und noch einige mehr) finden sich auf einer C 64-Sammeldiskette. Wer Interesse daran hat, kann diese Diskette für 10 Mark von der B.H.P. bekommen (Adresse und Hinweis »C 64 Sammeldiskette« angeben).

Datex-P — die preisgünstige Alternative

Bei diesem Wort bekommen die meisten Hacker leuchtende Augen, ist für sie Datex-P doch gleichbedeutend mit dem Tor zur Welt. Datex-P ist ein spezielles Netzwerk, das Computer aus der ganzen Welt verbindet.

Das Datex-P-Netz wurde 1980 von der Deutschen Bundespost eingerichtet, um der immer wichtiger werdenden Datenkommunikation Rechnung zu tragen. Bedingt durch die spezielle Art der Datenübertragung (sogenannte Paketvermittlung, daher auch das P hinter Datex) ist die Übertragung hier wesentlich günstiger als über das Telefonnetz. Darüber hinaus nimmt Datex-P notwendige Anpassungen der Computer untereinander selbst vor, wenn zum Beispiel ein Datenbank-Computer mit einer schnellen Schnittstelle mit 9600 Baud ausgerüstet ist, kann man dennoch mit dem derzeit noch üblichen 300-Baud-Koppler mit diesem Computer kommunizieren, da Datex-P die erforderliche Umsetzung selbst vornimmt.

Was braucht man, um Datex-P benutzen zu können?

1. Die normale Ausrüstung eines Hackers: Computer, Akustikkoppler und Terminalprogramm sowie Telefon.
2. Zusätzlich eine Benutzerkennung für Datex-P, eine NUI (Network User Identification).
3. Die Nummer des gewünschten Computers im Datex-P-Netz, die NUA (Network User Adress).

Die NUI kann man bei der Post beantragen. Man bezahlt, ähnlich wie beim Telefon, eine Grundgebühr von 15 Mark plus die monatlich anfallenden Kosten, die aber nur ungefähr ein Drittel dessen betragen, was man im normalen Telefonnetz bezahlt. Ein Tip: Wenn sich mehrere Freunde zusammentun und gemeinsam ihre NUIs bestellen, so kostet die Grundgebühr für die zweite und alle folgenden NUIs nur noch 5 Mark.

Nach unseren Erfahrungen ist es aber noch nicht bei allen Postämtern beziehungsweise Telefonläden bekannt, daß es Datex-P überhaupt gibt. Was man braucht, ist ein »Antrag für die Zuteilung einer Teilnehmerkennung Datex-P« und nichts anderes.

Wenn diese bürokratischen Hürden endlich genommen wurden, kann es losgehen:

Computer einschalten, Terminalprogramm laden, Telefonhö-

rer zur Hand und den nächsten Netzknoten Computer (PAD — Packet Assembly/Disassembly Facility) angerufen. Eine Liste der 300-Baud-Rufnummern findet sich am Ende dieses Artikels. Die vollständige Liste, in der auch die Anschlüsse für 1200 und 1200/75 Baud angegeben sind, findet man in diversem Werbematerial der Post.

Wenn es aus der Leitung piepst, Hörer in den Koppler legen und

<CR>

eingeben. (Punkt eingeben und die RETURN-Taste drücken).

Datex-P meldet sich daraufhin mit: Datex-P:

44 8900 49130.

Jetzt stehen maximal zwei Minuten zur Verfügung, um den selbstgewählten Teil A der NUI einzugeben. Also zum Beispiel:

NUI BEISPIEL <CR>

Der PAD antwortet:

Datex-P: Passwort:

XXXXXX

Hier gibt man dann den Teil B der NUI, also das Paßwort, das man per Einschreiben von der Post erhalten hat, ein. Aus Sicherheitsgründen erscheint der Teil B nicht auf dem Bildschirm.

Der PAD antwortet:

Datex-P: Teilnehmerkennung DBEISPIEL aktiv

Ab jetzt hat der Benutzer 60 Sekunden, um die Nummer des gewünschten Host-Computers einzutippen und eine Verbindung herzustellen:

NUA 45228040187 <CR>

Ist der Anschluß frei, so antwortet der PAD mit:

Datex-P: Verbindung hergestellt mit 45 2280 40187

Ab jetzt gehen alle Eingaben an den angewählten Host-Computer. Also zum Beispiel in diesem Fall an den Computer des Amtsgerichts Bonn.

Fortsetzung folgt.

Telefonnummern der PAD-Computer

Augsburg	0821/36791
Berlin	030/240001
Bielefeld	0521/59011
Bremen	0421/310131
Dortmund	0231/57011
Düsseldorf	0211/329318
Essen	0201/787051
Frankfurt	069/20281
Hamburg	040/441231
Hannover	0511/326651
Karlsruhe	0721/60241
Köln	0221/2911
Mannheim	0621/39931
München	089/228730
Nürnberg	0911/20571
Saarbrücken	0681/810011
Stuttgart	0711/299171

Über diese Nummern erhalten Sie Zugang zum PAD-Computer

Hilferuf aus einer Mailbox

Mein Name ist Michael Busboom. Ich beschäftige mich derzeit mit der Entwicklung von Hilfsmitteln für Blinde. Wenn ich folgendes Problem schildere, spreche ich von Herzen, denn ich bin auch blind, und meine Aussagen kommen nicht von ungefähr.

Derzeit gibt es einige blinde Kinder, die in völlig normale Regelschulen integriert werden. Die Integration bringt diesen Kindern enorme Vorteile, denn der Umgang mit sehenden Menschen bereitet sie auf ein Leben vor, in dem sie neue Berufsmöglichkeiten und Freiheiten haben werden. Zusätzlich wird das allgemeine Verständnis zwischen Blinden und Sehenden gefördert. Bisheriger Erfolg:

Da es notwendig ist, dem blinden Schüler Zugang zu den im Unterricht verwendeten Büchern zu verschaffen, wurden mittels eines ausgereiften technischen Verfahrens einige Bücher mit der Unterstützung eines Großcomputers in Blindenschrift übertragen. Andere Bücher wurden von einer kommerziellen Lesemaschine gelesen. Es gibt jedoch noch Bücher, die sich mit diesen Verfahren nicht verarbeiten lassen.

Diese Bücher müssen von Hand in einen Computer eingetippt werden. Können Sie helfen?

Mit der modernen Mikro-Elektronik ist es tatsächlich möglich, Texte in Blindenschrift zu erstellen, ohne über besondere Kenntnisse zu verfügen. Wenn jeder Mailbox-Benutzer sich bereit erklärte, nur einige Seiten einzutippen, so wäre der Bedarf an noch benötigten Büchern völlig gedeckt. Wie können Sie uns helfen, dieses Problem zu bewältigen? Es ist uns derzeit nicht möglich, ein Honorar für diese Arbeit zu zahlen. Die Arbeit ist daher als caritativ zu betrachten. Wenn Sie bereit sind, uns zu helfen, so ersuchen wir Sie, uns eine kurze Mitteilung zu schicken. Anschließend erhalten Sie weitere Hinweise.

Mit der Hoffnung, dieses dringende Problem gemeinsam lösen zu können:

Michael Busboom
IMCA + RMI
Mailbox-Adresse: Vielhaber

Chaos Communication Congress (CCC) 1985

Ende Dezember 1985 findet in Hamburg zum zweitenmal der Chaos Communication Congress, ein Treffen von Hackern aus der ganzen Welt, statt. Veranstalter ist der aus Rundfunk- und Fernsehwerbung bekannte Chaos Computer Club aus Hamburg. Auf dem Programm stehen auch dieses Jahr wieder Vorträge und Diskussionen rund um Telefon und Datennetze, Vorführungen und Berichte aus der Welt der Hacker.

Der genaue Termin stand bei Redaktionsschluß noch nicht fest, wahrscheinlich aber Samstag/Sonntag, 27./28. Dezember.

Handbuch für Hacker und andere Freaks

Das absolute Muß für jeden Hacker und die, die es werden wollen. Bisheriger Inhalt: Kostenlos telefonieren, Viren, was ist eine Mailbox, Mailboxnummern im In- und Ausland, Datex-P und seine Wanzen, B.H.P., DS/CCC, NUA-Sammlung, VATICAL (TM), Hardwaretips und vieles mehr. Bisheriger Inhalt deshalb, weil das HfH eine lose Blattsammlung darstellt, die in einem DIN A5-Ordner vertrieben wird. Um bleibende Aktualität zu sichern, wird das Buch durch drei im Preis enthaltene Nachlieferungen ergänzt. Bis heute (03. 11.85) sind schon zwei beim Rezensor eingetroffen. Die dritte soll zu Weihnachten 1985 fertig sein. Ab Anfang 1986 erscheint dann ein vierteljährlicher Infobrief im selben Format. Außerdem werden Infofiles über verschiedene Mailboxen in ganz Deutschland abrufbar sein.

Handbuch für Hacker
Rathmann — Schalla
Friesenstr. 24
3000 Hannover 1
Preis: 20 Mark für vier Ausgaben.

Datex-P-Nummern

Univ. Nürnberg	45615140282
Telebox	45621040000
Univ. of Essex	2342206411411
Decates	44615440307
Lockheed Dialog	31104150002000
Univ. of Alberta	302060100010
Nat. Lib. of Medicine	3106005 0
Euronet Diane	2342192010100513
American Express	44611040303
ITT	23422790014302
Univ. of Wales	234222236236
Ferranti	234261643343
Cyber Erlangen	45913111
Primnet	R45211040026
USA-Telenet	311020200142 75
Dartmouth College	311060300020
ESA	2223077,Quest

Hackerbibel

Seit langem erwartet und doch fertig geworden. Am 16. September 1985 sind die ersten Exemplare der Hackerbibel des CCC ausgeliefert worden. Der Inhalt der 260 Seiten ist zu vielfältig, um in einigen Worten beschrieben zu werden. Außer einer Reihe von kaum bekannten Artikeln aus dem In- und Ausland, sind alle bisher erschienenen Ausgaben der Datenschleuder und eine Auswahl aus den ersten 27 TAP's enthalten.

Die Hackerbibel
ISBN3-922708-98-6
Verlag Werner Pieper
D-6941 Löhrbach
Preis: 33 Mark

(B.H.P./aw)

Sind Sie ein Geheimniskrämer?

Wenn nicht, dann haben wir ein interessantes Angebot für Sie. Gefragt sind Ihre Erfahrungen mit allen Programmen, Hardware-Erweiterungen und Druckern, die es für den C 64/C 128 gibt.

Man kann ein Computersystem auf zwei verschiedene Arten interpretieren. Zum einen ist das eine mehr oder weniger umfangreiche Ansammlung der verschiedensten Programme, Peripheriegeräte und Schnittstellen. Zum anderen ist ein Computersystem also eine Einheit, bei der es darauf ankommt, daß alle Teile des Systems aufeinander eingespielt sind und zueinander passen. Die vielen Briefe, die uns täglich erreichen, zeigen, daß dieser Idealzustand leider noch längst nicht erreicht ist. Immer wieder kommt es vor, daß ein Programm, beispielsweise eine Grafikerweiterung oder eine Textverarbeitung, nicht mit dem favorisierten Drucker mangels geeignetem Interface harmonisiert. Bekannte Problemkinder sind auch die immer beliebter werdenden Floppy-Speeder, die zwar mit vielen Programmen funktionieren, aber eben nicht mit allen. Diesen Zustand wollen wir ändern. Schreiben Sie uns Ihre Erfahrungen mit den verschiedensten Teilen Ihres Computersystems (bitte genau spezifizieren). Besonders interessant für uns ist natürlich, wie Sie trotz Schwierigkeiten zu guten Ergebnissen gekommen sind. Manchmal ist es nur eine Kleinigkeit, die es ermöglicht, daß die gewünschten Resultate erlangt werden. So lassen sich beispielsweise beim Data Becker-Centronics-Interface (das von Wiesemann gefertigt wird und damit die gleichen Befehle hat) viele Anpassungsprobleme lösen, wenn man das Interface zunächst in den Linearkanal schaltet und danach fixiert. So lassen sich viele Grafik-

und Textprogramme problemlos, mit allen ihren Funktionen, verwenden. Sie sehen, ein Tip muß nicht unbedingt umfangreich sein — Wirksamkeit ist gefragt.

Obwohl wir in unserer Redaktion ein wahrlich reichhaltiges Sortiment der verschiedensten Programme und Peripheriegeräte zur Verfügung haben, ist es natürlich auch uns nicht möglich, jedes Computersystem aufzubauen und alle Möglichkeiten auszuprobieren. Diese Möglichkeit haben nur Sie — unsere Leser. Warum mit diesen Informationen »hinter dem Berg halten«? Schreiben Sie uns Ihre Erfahrungen und Problemlösungen — es lohnt sich! Unter allen Zuschriften werden wir einen Überraschungspreis vergeben, für den es sich lohnt, die eigene Trickkiste etwas zu öffnen. Schicken Sie Ihren Tip unter Angabe der genauen Produktbezeichnung einschließlich der Versionsnummer (falls vorhanden) unter dem Stichwort »Computer Systeme« an:
Markt & Technik Verlag Aktiengesellschaft,
Redaktion 64'er,
Arnd Wängler,
Hans-Pinsel-Str. 2,
8013 Haar bei München

Mangel am Cobra-Joystick behoben

Das in unserem Test in der Ausgabe 11/85 bemängelte Abreißen der Saugnäpfe wurde von der Firma Rushware beseitigt. Alle Lieferungen seit Anfang Oktober beinhalten schon die veränderten Joysticks. Die scharfkantigen Lochränder wurden aufgebohrt und abgerundet. Zugleich sind die Saugfüße erneuert worden.

Info: Rushware GmbH, An der Gumpgesbrücke 24, 4044 Kaarst 2

Joystickkompatible Maus

Von der Firma Rushware wird jetzt eine Maus angeboten, die auf allen gängigen Heimcomputern lauffähig ist. Dazu wird keine besondere Treibersoftware benötigt, die Maus kann anstelle des Joysticks in jedem Programm, das für einen Joystick konzipiert wurde, verwendet werden. Die Maus ist mit zwei gleichberechtigten Tasten ausgerüstet, und somit auch für Linkshänder geeignet. Die Unterseite wurde mit Rollkugeln versehen, um auf jeder Unterlage arbeiten zu können. Der Verkaufspreis wird bei etwa 180 Mark liegen.

Info: Rushware GmbH, An der Gumpgesbrücke 24, 4044 Kaarst 2

Neuer Joystick — Quickshot IX

Unter dem Namen »Joyball« wird von Jöllenbeck der neue Quickshot IX angeboten. Die Halbkugel des Joyballs dient wie der Griff eines Joysticks der Richtungsänderung. Der Joyball ist mit Mikroschaltern, zwei Feuerknöpfen und automatischem Dauerfeuer ausgestattet. Der Verkaufspreis wird bei 59 Mark liegen.

Info: Bernd Jöllenbeck GmbH, Postfach, 2730 Weertzen

Preisrutsch

Es gehört (nicht immer zur Freude der Hersteller) in der Computerbranche schon fast zum guten Stil, daß der Preis nach einer Einführungsphase langsam aber stetig sinkt. Die Entwicklung, die der Preis des in Ausgabe 6/1985 getestete Typenraddrucker Uchida DWX 305 derzeit erlebt, ist aber außergewöhnlich. Der Drucker, der in unserem Test zwar nicht zu den schnellsten (18 Zeichen pro Sekunde, 32 Zeichen/s über alle Funktionen) gehörte, dafür aber mit einer sehr soliden Mechanik und der bei Typenraddruckern üblichen Schriftqualität überzeugte, hat seinen Preis schlichtweg um mehr als die Hälfte verringert. Er kostet jetzt mit Friktionsantrieb und Centronics-Schnittstelle 599 Mark, zu denen noch die Anschaffung eines Traktors für 275 Mark empfehlenswert ist. Selten gab es so viel Drucker für so wenig Geld. (aw)

Info: Weber Computertechnik, Ludmillastr. 15, 8000 München, Tel. (089) 6516856

Neues vom NEC P2

Der 18-Nadel-Drucker NEC P2, der in unserer letzten Ausgabe beinahe das Reverenzgerät FX-85 abgelöst hätte, hat dazugelernt. Er besitzt jetzt neben der bisherigen Schnittstelle mit der Bestellnummer 6302 eine weitere mit der Bezeichnung 6303. Der Vorteil liegt darin, daß der P2 dadurch etwa 98 Prozent der FX-85-Befehle versteht. Gleichzeitig läßt er sich mit DIL-Schaltern auf die IBM-Zeichensätze eins und zwei einstellen. Der Preis ist gleich geblieben. Die im Testbericht vergrößert abgebildete NLQ-Schrift läßt sich übrigens nochmals verbessern; wählt man statt der Schriftart eins die Schriftart zwei, treten kaum noch Einzelpunkte hervor. (aw)

Info: NEC-GmbH, Klausenburger Str. 4, 8000 München 80

Messen — Steuern — Regeln

Die Arbeitsgemeinschaft Solartechnik Bergstr. e.V. führen an verschiedenen Volkshochschulen Kurse zum Thema Messen—Steuern—Regeln durch. Ziel ist es, eine Einführung in Meß-, Steuer- und Regelvorgänge im Haushalt zu geben. Am Beispiel eines C 64 werden die notwendigen Programme und die erforderliche Hardware besprochen und erstellt. Zur Demonstration allgemeiner Vorgänge dient ein Fischer-Technik-Roboter, eine Eisenbahnsteuerung und eine Heizungssteuerung. Für den Kurs sollen Elektronikkenntnisse nicht erforderlich sein. Jedoch sollte man Basic-Kenntnisse mitbringen. Die Kursdauer beträgt 8 Tage zu je 3 Stunden. Weitere Informationen erhalten Sie gegen 1,30 Mark Rückporto in Briefmarken von der Kontaktstelle
AG Solartechnik Bergstraße e.V.,
W. Konkol,
Haydnstr. 4,
6909 Mühlhausen (hm)

Gut aufgeräumt

Ein »aufgeräumtes« Computersystem soll der Computer-Caddy (Bild) von ECO schaffen. In einem Schrank finden alle Peripheriegeräte und Utensilien eines Heimcomputers oder PC Platz. Auch das Kabelgewirr soll mit dem Computer-Caddy ein Ende haben. Den Computer-Caddy gibt es in Schleiflack und Funier. (hm)

Info: ABL-Service, Postfach 1151, 6240 Königstein, Tel. 06174/7070

Der Computer-Caddy soll Ordnung schaffen ▼



Neues von Rushware

Gleich mehrere Neuigkeiten sind von der Software-Vertriebsfirma Rushware zu vermelden.

So ist jetzt exklusiv bei Rushware eine Spezial-Version des James-Bond-Spiels »A View to a Kill« erhältlich. Auf der B-Seite der Kassette befindet sich anstelle von Computer-Software eine Cover-Version des Single-Hits von Duran Duran. Diese Disco-Version hatte am 23.09.1985 schon Platz 24 der deutschen Single-Hitparade erreicht.

Rushware hat einen Vertrag mit dem amerikanischen Hersteller Epyx geschlossen, der Rushware die deutschen Exklusivrechte für neue Epyx-Produkte sichert. So werden in diesen Tagen die Spiele »Winter Games« und »Temple of Apshai Trilogy« in deutscher Verpackung und mit deutschsprachiger Anleitung ausgeliefert.

Zusätzlich hat Rushware ein neues Software-Label, »Eurogold«, gegründet. Alle »Eurogold«-Spiele werden 9,95 Mark kosten. Unter den über dreißig Titeln, mit denen »Eurogold« gestartet wird, befinden sich einige Epyx-Klassiker (Jumpman, Pitstop I), aber auch brandneue Programme von verschiedenen Programmierern aus ganz Europa und Amerika.

Als letztes ist noch die Gründung der Rushware-Mailbox zu vermelden. Unter der Nummer 02101/66778 soll man unter anderem neue Produktinfos für Zubehör und Software abrufen. Es handelt sich dabei um eine Standard-Mailbox mit 300-Baud-Übertragungsrate. Sie bietet Up- und Download-Funktionen. Die Mailbox läuft 24 Stunden am Tag und der Sysop, Heinz Fischer, soll fast jeden Abend zu erreichen sein. (bs)

Info: Rushware, An der Gümpgesbrücke 24, 4044 Kaarst 2





Fließkommaroutinen beim VC 20

Besitzt der VC 20 eingebaute Maschinensprache-Routinen für Multiplikation und Division?
Ausgabe 11/85
Manfred Klemenz

Der VC 20 besitzt eine Maschinensprache-Routine zum Multiplizieren; die Startadresse ist $D357, desgleichen eine Divisionsroutine. Die Zero-Page-Adressen, die für diese Routinen verwendet werden, sind dieselben wie beim C 64 und wurden in der Ausgabe 5/85 des 64'er-Magazins im Assemblerkurs besprochen. Die in diesem Kursteil ebenfalls beschriebene Blockverschieberoutine startet beim VC 20 auf der Adresse $C3BF, die dazu gehörigen Speicherstellen sind wiederum dieselben wie beim C 64.

Meindert Hielkema, Niederlande

Fragen Sie doch

Selbst bei sorgfältiger Lektüre von Handbüchern und Programmbeschreibungen bleiben beim Anwender immer wieder Fragen offen. Viel mehr Fragen ergeben sich bei Computer-Interessenten, die noch keine festen Kontakte zu Händlern, Herstellern oder Computerclubs haben. Sie können der Redaktion Ihre Fragen schreiben oder Probleme schildern (am einfachsten auf der Karte »Lesermeinung«). Wir veranlassen, daß sie von einem Fachmann beantwortet werden. Allgemein interessierende Fragen und Antworten werden veröffentlicht, die übrigen brieflich beantwortet.

Probleme mit MPS 801 und Floppy 1541?

Als Besitzer eines VC 20 mit 32/27 KByte-Modul habe ich nie Probleme gehabt, mit meinem MPS 801 Programme auszudrucken, die zuvor per Datasette gelesen wurden. Das hat sich schlagartig geändert, seit ich im Besitz einer 1541-Floppy bin. Beide Geräte, Drucker und Floppy, sind ordnungsgemäß angeschlossen und arbeiten für sich alleine auch einwandfrei. Soll allerdings ein Programm bei angeschlossener Floppy ausgedruckt werden, dann führt das regelmäßig zum »Absturz« des Computers. Auch Entfernen aller Erweiterungen führte zu keiner Verbesserung. Sobald sowohl Drucker als auch Floppy gleichzeitig angeschlossen sind, kann ich den Drucker vergessen. Das kann doch wohl nicht im Sinne des Erfinders sein?! Ich wende mich an das Leserforum, weil das, was mir die Verkäufer der Geräte auf meine wohl berechtigten Reklamationen geboten haben, hart an die Grenze des Anstands gegangen ist.
Rudolf Inckemann

Computer für Blinde?

Gibt es ein Programm für den C 64 oder für einen anderen Computer, mit dem auch Blinde und stark Sehbehinderte die Möglichkeit bekommen, einigermaßen sinnvoll mit dem Computer arbeiten zu können? Ich denke dabei an ein Programm, das bei allen Bildschirmausgaben entweder jedes Zeichen im Klartext ansagt (Sprachsynthese) oder wenigstens jedem Zeichen einen bestimmten Ton beziehungsweise eine Tonfolge zuordnet.
Gerhard Halmerbauer

Haushaltsbuch mit MPS 801

Wie kann man das Programm »Haushaltsbuch« aus der Ausgabe 7/85 für den MPS 801 umschreiben?
Ausgabe 11/85
Dirk Zwartenbol, Niederlande

Ich habe dieses Problem gelöst durch Ändern der Zeilen 250, 2830, 5890, 5910, 5930, 5960, 5970. Zeile 5880 habe ich fortgelassen. Hier die für den Betrieb mit MPS 801 geänderten Zeilen:

```
250 POKE 768,185 : OPEN 1,4,7 : PRINT#1,"" : CLOSE 1 : POKE 768,F%
2830 CLOSE 1 : OPEN 4,4,7 : PRINT#4 : CMD 4 : IF X=3 THEN 2540
5890 PRINT#1, D$(Y,0); SPC(14); D$(Y,1); SPC(8)
5910 PRINT#1, MID$(D$(Y,2),3,2); "."; LEFT$(D$(Y,2),2); "."; RIGHT$(D$(Y,2),2)
5930 PRINT#1, MID$(D$(Y,2),2,2); ". "; LEFT$(D$(Y,2),1); "."; RIGHT$(D$(Y,2),2)
5960 PRINT#1, SPC(20); "--------" : PRINT#1, "SUMME "; : GOSUB 4370
5970 PRINT#1, SPC(19-(L-1)); SU$ : CLOSE 1 : A$="" : RETURN
```

Mit diesen Änderungen läuft das Haushaltsbuch auch mit dem MPS 801.

Sabine Gnutzmann

Turbo-Lader eingebaut

Wie muß das Betriebssystem-ROM beim C 64 geändert werden, um ein Turbo-Tape-Programm fest zu integrieren? Welche Adressen bestimmen die Baudrate?
Ausgabe 11/85
Bernt-Dieter Bernt

Die Baudrate des Kernels zu erhöhen reicht alleine nicht aus, um auf die Geschwindigkeit von Turbo-Tape zu kommen. Wir haben das Betriebssystem dahingehend geändert, daß alle Programme mit der Geschwindigkeit von Turbo-Tape geladen und gespeichert werden können.

K. Hallmann, Christian Gallas

Ich habe ein Turbo-Tape-Programm ins Betriebssystem eingebunden, das als Besonderheit auch solche Programme bearbeiten kann, die bis maximal $D000 (also bis unter das Basic-ROM) reichen. Ein entsprechendes EPROM, das anstelle des normalen Betriebssystems eingesteckt wird, kann ich ebenso wie einen notwendigen Adaptersockel zur Verfügung stellen.

Franz Hirschmann

»Kurvenplotter C 16« für Star-Drucker?

Wie und mit welcher Anpassung läuft das auf dem Bildschirm ganz brauchbare Programm zur Kurvendarstellung, so daß es eine Hardcopy auch auf meinem Star SG 10 mit Star-Interface ausdruckt? Mein Plus 4 meldet lediglich »?File Open Error«, wenn ich das Programm ohne Änderungen starte.
Günter Dewald

Kurzwelle und C 64

Gibt es die Möglichkeit, Kurzwellenfunksendungen mit dem C 64 zu verarbeiten?
Ausgabe 11/85
Holger Jöhnck

Ich bin selbst lizenzierter Radioamateur und verwende den C 64 im Zusammenhang mit KW-Empfang und RTTY (Radio-Teletype). Was brauchen Sie dazu?

Hardware:
- Eine gute KW-Außenantenne
- Einen Kurzwellenempfänger mit Beat-Oszillator
- Ein Interface, welches die ankommenden analogen Signale in digitale Daten umwandelt
- Software: Ein spezielles Programm für KW-Verarbeitung, was meistens auch RTTY beinhaltet.

Für weitere Informationen können Sie sich an den DARC (Deutscher Amateurradio Club) wenden, der Sie sicher weiter beraten wird.

Hinsichtlich Interface und Software müßte Ihnen die Firma P. Walter in Celle weiterhelfen können.

Ich würde mich freuen, wenn Sie eines Tages mein Signal (SM0KWK) auf Ihrem Monitor sehen würden . . .

Holger Kossmann, Schweden

DARC, Postfach 1155, 3507 Braunatal 1, Firma P. Walter, An der Ziegelei 1, 3100 Celle

C 64 und Video-Recorder

Wer hat Erfahrung mit Einsatz und Programmierung eines C 64 als Steuereinheit für einen Video-Recorder? Ich möchte zwei Video-Recorder über den User-Port ansteuern können. Als Taktsignal steht der CTL-Impuls mit 25 Hz zur Verfügung. Über die Ausgänge des User-Ports möchte ich die Funktionen START, STOP, PAUSE etc. ansteuern. Das Taktsignal soll Anfang und Ende einer Steuersequenz bestimmen. Wer hat hierzu Ideen und eigene Erfahrungen?
Günter Kelleter

Hi-Eddi und 1526

Wie bekomme ich eine Hardcopy von Hi-Eddi-Bildern mit dem 1526-Drucker zustande?

Heinz Krieger
Ausgabe 11/85

Mit dem Drucker 1526 und der Betriebssystemversion REV 07 ist ein Grafik-Ausdruck nicht möglich. Ich hatte das gleiche Problem und habe kurzerhand das Betriebssystem-EPROM des Druckers gelöscht und mit dem MPS 802-Betriebssystem neu programmiert. Danach war ein einwandfreier Grafikbetrieb möglich. Auf Wunsch programmiere ich Herrn Kriegers EPROM gern um.

Bernhard Korell

C 64 als Btx-Tastatur

Kann man den Commodore 64 auch als Btx-Tastatur benutzen? Ausgabe 11/85

Jörg Stalberg

Der C 64 läßt sich als Btx-Tastatur einsetzen, wenn ein Btx-Gerät vorhanden ist. Ich habe ein entsprechendes Programm selbst geschrieben. Es hat die gleichen Funktionen wie eine echte Btx-Tastatur. Zusätzlich können Btx-Seiten auf Diskette gespeichert und wieder geladen werden. Das Programm läuft allerdings nur mit Loewe-Decodern, wie sie zum Beispiel auch in Philips-Geräten eingebaut sind. Weiterhin gibt es eine ganze Reihe von Firmen, die ein solches Programm anbieten, zum Beispiel Commodore.

Bernhard Korell

Fragen zum C 16/Plus 4

(1) Ist es möglich, mit dem C 16 beziehungsweise dem Plus 4 Datenfernübertragung zu betreiben, und was benötigt man dazu?

(2) Ist es möglich, von der Post ein Modem zu bekommen, das auch für den Plus 4 geeignet ist, und welches Modem ist das beste, um zum Beispiel mit Mailboxen Datenfernübertragung zu betreiben?

(3) Warum wurde bei den neuen Commodore-Computern C 16, 116 und Plus 4 auf den User-Port verzichtet? Welchen Grund hatte Commodore dafür; es ist ja schon ärgerlich genug, daß die Anschlüsse für Datasette und Joysticks geändert wurden.

Peter Ludwig

(1) Zur Datenfernübertragung mit dem Plus 4 brauchen Sie eine serielle Schnittstelle (RS232) und entsprechende Treibersoftware für diese Schnittstelle. Mit einer RS232-Schnittstelle können Sie jeden Akustikkoppler anschließen, desgleichen praktisch jedes Modem. Doch damit sind wir bereits bei einem großen Haken an der ganzen Geschichte: Zum Betrieb eines Akustikkopplers braucht man ein Terminalprogramm. Für weit verbreitete Computer wie den C 64 oder die Schneider Computer ist das kein Problem, da gibt es jede Menge Auswahl. Beim Plus 4 wird's da leider schon etwas kritischer. Möglicherweise hilft da nur selber schreiben. Vielleicht kann da aber auch einer unserer Leser weiterhelfen? Über serielle Schnittstellen zum Plus 4 informiert Sie auf jeden Fall Ihr Commodore-Händler.

(2) Ein Modem wird nicht direkt mit dem Computer angeschlossen, sondern an die serielle (RS232) Schnittstelle. Damit ist die Wahl des Modems nicht an einen bestimmten Computertyp gebunden. Über zugelassene Modems informiert Sie die Deutsche Bundespost.

(3) Wenn es Ihnen ein Trost ist: Das fragen wir uns auch.

C 16 statt C 64?

Ich besitze seit geraumer Zeit einen C 64 nebst Peripherie. Für manche Anwendungen reicht der Speicher des C 64 aber nicht aus. Schon 10 KByte mehr von Basic aus ansprechbar würden es aber tun. Darum möchte ich mir nun einen C 16 mit 64 KByte Speichererweiterung zulegen. Erstens besitzt dieser Computer ein weitaus besseres Basic und zweitens kann ich Floppy, Drucker und Plotter des C 64 weiterverwenden.

Dazu habe ich folgende Frage: Ist der C 16 bei dieser vollkommenen Speichererweiterung voll zum C 16 in der Grundausführung kompatibel?

Joachim Eiselt

Leider gibt es zum C 16 von Commodore selbst keine Speichererweiterung, so daß man nur Fremdfabrikate verwenden kann. Der C 16 ist jedoch vom Betriebssystem her bereits auf einen Speicher von 64 KByte für Basic-Programme ausgelegt. Von der Basic-Ebene her betrachtet ist der C 16 bei Vollausbau kompatibel zur Grundversion, desgleichen zum Plus 4, der ja bereits serienmäßig über 64 KByte Speicher verfügt.

Leider hört die Kompatibilität bei PEEK, POKE und SYS auf. Bei Programmen, die diese Befehle benutzen, kann ein einwandfreies Funktionieren nicht in jedem Fall garantiert werden. Die Änderungen am Programm, die im Falle eines Falles gemacht werden müssen, sind aber in der Regel minimal.

Allerdings sollten Sie sich Ihren Entschluß vielleicht doch nochmals überdenken: Der C 16 ist, wie leider auch der Plus 4, völlig inkompatibel zum C 64, so daß Sie große Probleme bekommen werden, wenn Sie versuchen, Ihre C 64-Programme für den C 16 »umzustricken«, insbesondere, wenn die Programme mit Maschinensprache-Unterroutinen arbeiten.

Zudem gibt es für den C 16 so gut wie keine Software zu kaufen, erst recht keine, die irgendwelche Speichererweiterungen ausnutzen kann. Sie müssen sich darüber im klaren sein, daß Sie mit einem C 16 und Speichererweiterung einen Computer besitzen, der höchstens kompatibel zu sich selbst, aber zu nichts anderem ist.

Schließlich sollten Sie sich überlegen, ob sich Ihr Problem nicht auch durch effizientere Programmierung oder bessere Ausnutzung des Floppy-Laufwerks lösen läßt. Und dann gibt es ja auch noch das Business-Basic-Steckmodul von Kingsoft (Test in Ausgabe 8/85), das den Basic-Programmspeicher des C 64 auf 61 KByte RAM erweitert und nebenbei noch über 50 neue Befehle zur Verfügung stellt.

Fragen ohne Grenzen

Was braucht man alles, um ein professionelles Spiel (nicht so gut wie »Impossible Mission« und nicht so schlecht wie ein billiges Arcade-Game wie »Defender«) zu schreiben, wenn man den C 64 einigermaßen beherrscht, wie zum Beispiel, VIC, SID, CIA, Zeichensatz? Was ist wichtig zu wissen, um die Sprites auf dem Fernseher, hinter oder in einem scrollenden Grafik-Bildschirm zu bewegen? Ich scheue keine Hilfsmittel wie zum Beispiel einen Sprite-Monitor oder Pascal.

Ich möchte einen Kopierschutz für meine Programme. Ich will durch einen Hardware-Fehler auf der Diskette einen Ausstieg mit bestimmter Fehlermeldung erzwingen, sonst wird die Diskette formatiert und gelöscht.

Das sind zwei von vielen Fragen, die monatlich unsere Redaktion erreichen. Leider sind sie auch typisch für eine Gruppe von Fragen, die sich in letzter Zeit häuft. Fragen, die derart allgemein gehalten sind, können wir beim besten Willen weder im Leserforum noch schriftlich beantworten.

Wir können leider keine Kurse über professionelle Spielerprogramme bringen, denn es gibt aus guten Gründen nur wenige gute Spiele. Programmierer finden nur durch Probieren ihre Tricks heraus und behalten sie dann für sich. Außerdem braucht man einen guten Schuß Phantasie — Schließlich will man ja kein Allerweltsspiel programmieren sondern etwas Eigenständiges. Auch das Thema Kopierschutz wirft enorme Probleme auf: Die Veröffentlichung eines Schutzes macht diesen sofort wieder wirkungslos, da ja jeder Semi-Knacker ihn ganz genau kennt. Und Ihnen persönlich einen Kopierschutz zu entwickeln, dazu fehlt uns leider die Zeit, denn wir wollen ja jeden Monat eine interessante Zeitschrift machen. In dieser Zeitschrift, der 64'er, und in unseren Sonderheften können Sie ja fast immer Kurse und Artikel über bestimmte Programmier-Techniken finden, vom Abenteuerspiel bis zum Stringsortieren. Allerdings können wir solche Artikel nicht für alle möglichen Bereiche auf Abruf unserer Leser bereithalten, denn sie müssen erst einmal geschrieben werden.

Also eine Bitte — Wir sind jederzeit für Fragen und Probleme unserer Leser offen, die wir beantworten können, ohne dafür mehrere Arbeitstage an einer Antwort zu sitzen. Lösungen für ganz allgemeine Fragen können wir Ihnen leider nicht bieten.

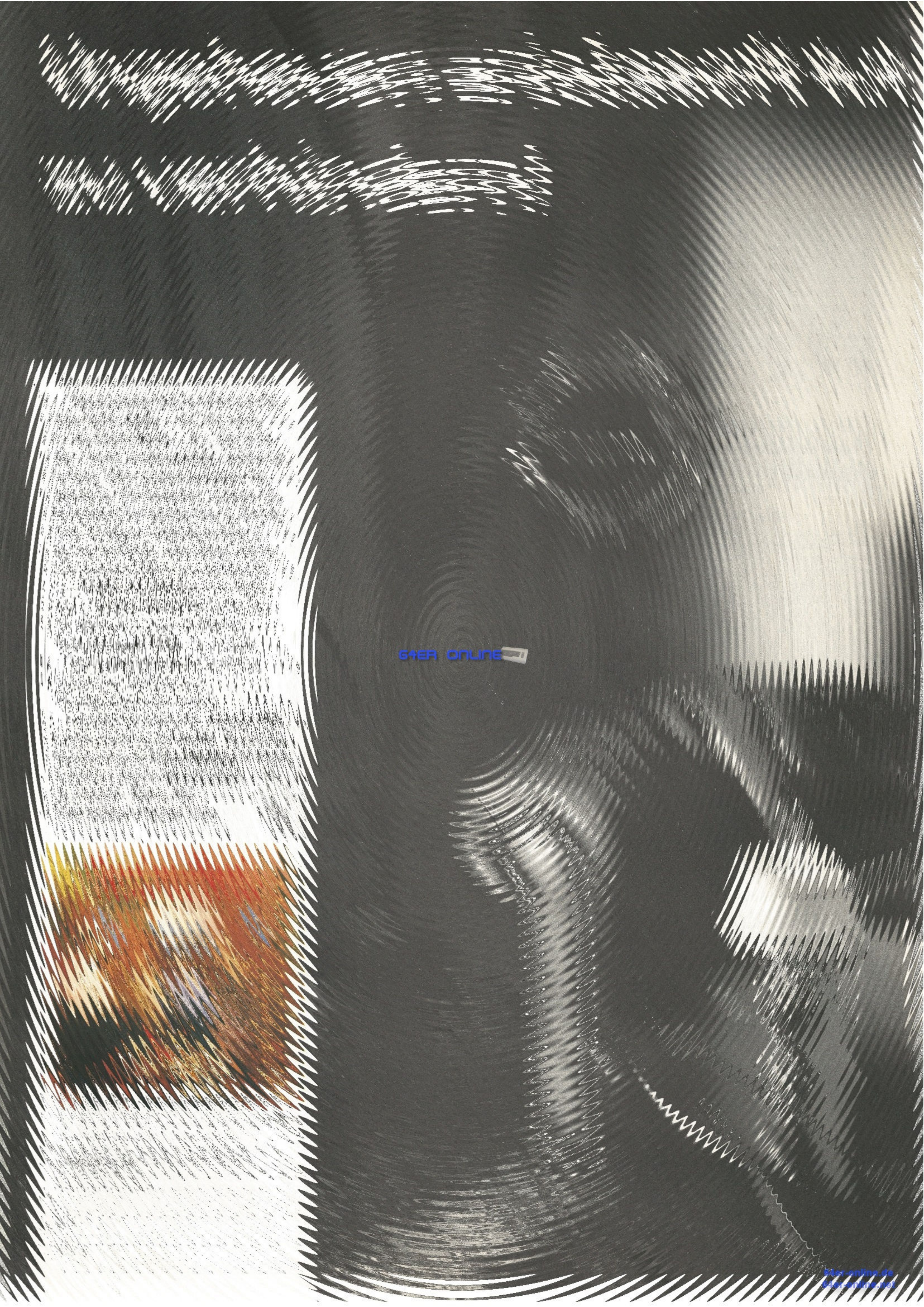
Directory mit MPS 802?

Wie kann ich mit einem MPS 802-Drucker das Inhaltsverzeichnis meiner Disketten in der Mitte des Papiers beziehungsweise am rechten Rand ausdrucken? Natürlich ist mir die im Handbuch beschriebene Methode mit »OPEN 1,4 : CMD 1 : LIST« bekannt, doch erscheint dabei das Directory immer am linken Rand.

Uwe Melchiar

Wollen Sie antworten?

Wir veröffentlichen auf dieser Seite auch Fragen, die sich nicht ohne weiteres anhand eines guten Archivs oder aufgrund der Sachkunde eines Herstellers beziehungsweise Programmierers beantworten lassen. Das ist vor allem der Fall, wenn es um bestimmte Erfahrungen geht oder um die Suche nach speziellen Programmen. Wenn Sie eine Antwort auf eine hier veröffentlichte Frage wissen — oder eine andere, bessere Antwort als die hier gelesene, dann schreiben Sie uns. Antworten publizieren wir in einer der nächsten Ausgaben. Bei Bedarf stellen wir auch den Kontakt zwischen Lesern her.





Riteman II – der Originalgetreue

Der Riteman II ist ein Drucker für überquellende Schreibtische. Er überrascht mit kleinen Abmessungen. Äußerlich klein, bietet er aber fast das gleiche wie sein Vorbild, der FX-85.

Bei der ersten Betrachtung des Riteman II (Bild 1) verblüfft das kompakte, aber schwere Gehäuse, das kleiner als das eines MPS 802 ist. Die Tasten SEL (entspricht On-/Offline), LF (Zeilenvorschub) und FF (Seitenvorschub) sind an der gleichen Stelle wie beim Vorbild angebracht. Leider ist die Möglichkeit der Auswahl der Schriften mittels dieser Tasten beim Riteman II nicht implementiert. Auf der Rückseite findet man, nach Entfernen einer kleinen Abdeckung (ohne Werkzeug), 16 DIP-Schalter zur Auswahl verschiedener Einstellungen. Um nur einige wichtige zu nennen:

Es stehen acht internationale Zeichensätze zur Verfügung. Die Null kann mit und ohne Schrägstrich gedruckt werden. Man hat auch die Wahl, ob sich der Riteman II nach dem Einschalten im NLQ- oder im Normalmodus befinden soll. Papierlänge und die Abschaltung des Warnsignals werden an diesen Schaltern ebenfalls eingestellt. Der Riteman II wird serienmäßig mit einer Centronics-Schnittstelle und 2-KByte-Puffer ausgeliefert. Als Option wird ein 8-KByte-Puffer (215 Mark) und eine serielle Schnittstelle (RS232C, Preis: 130 Mark) angeboten. Der Druckkopf macht einen soliden Eindruck und ist auf zwei Führungsschienen montiert. Dies sind gute Voraussetzungen für ein sauberes Schriftbild. Man darf wohl der gesamten Mechanik das Prädikat »solide und funktionell« verleihen.

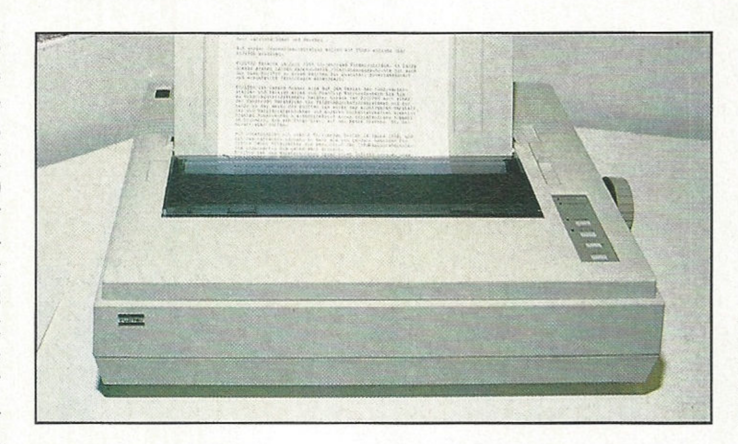


Bild 1. Der Riteman II – eine geballte Ladung Drucker

Tinte statt Carbon

Die Farbkassette (20 Mark) ist nur etwa sechs Zentimeter breit und wird mit dem Druckkopf mitgeführt. Das Wechseln der Kassette ist fast beispielhaft einfach und sauber, obwohl es sich um ein mit Tinte getränktes Band handelt. Die Kassette ist nachfüllbar und da das Band nahtlos als Endlosband ausgeführt ist, kann man auf eine lange Lebensdauer hoffen. Wie beim Vorbild FX-85 ist die Papierführung konstruiert: Die Stachelwalzen befinden sich auf derselben Achse wie die Andruckrolle. Das Einlegen von Endlos- und Einzelblättern geht mühelos und stellt keine hohen Ansprüche an die Fingerfertigkeit des Besitzers. Die ganze Mechanik wird von einer Plexiglasabdeckung gegen Staub und die Finger eines zu forschen Bedieners geschützt. Die im Handbuch als Abrißkante bezeichnete Schiene zum Abreißen des Papiers wird ihrem Namen mangels Schärfe allerdings nicht ganz gerecht. Endlospapier kann hier nur sehr schlecht abgerissen werden. Meistens reißt man das Papier dabei schräg ab. Ein Umstand, der sich sicher sofort und ohne großen Aufwand seitens des Herstellers ändern lassen müßte.

Kurz belichtet

Merkmal	Wert	Merkmal	Wert
Name des Druckers	Riteman II	empfohlener Preis	1398 Mark
Unterstreichen	Ja	Proportionalschr.	Ja
Zeichenmatrix	9 x 9	NLQ-Matrix	16 x 12
Papierbreiten	101,5 bis 254	Zeichenvorrat	ASCII, 8 Intern
Papierarten	Einzel/Endlos	Durchschläge	2
Zeichen pro Zeile	0 bis 131	Selbsttest	Ja
Hexdump	Ja	Autom. Einzelbl.	Nein
Pufferspeicher	2 bis 8 KByte	Rückwärtstransp.	Ja
Ladb. Zeichensatz	Ja	Probetext	1:50 Minuten
Geschwindigkeit	160, Messung 140	NLQ-Geschw.	24 Zeichen/s
Grafikmodi	bis 4fache Dichte (1920 Druckpositionen)		
Funktionstasten	SEL = ON/OFF-Line, LF, FF, 16 DIP-Schalter		
Ausstattung	deutsches Handbuch, Farbband		
Schriftarten	Pica, Elite, Schmal, Breit, Doppel, Fett, Italic		
Sonderfunktionen:	halbe Geschwindigkeit, NLQ-Schrift		

Tabelle. Viele Funktionen für wenig Geld: Riteman II



Bild 2. Der fünffach vergrößerte Buchstabe A in Schönschrift

Der RITEMAN II
Eliteschrift
Elite- und fett
Normalschrift
Schmalschrift
breit
Doppeldruck
hoch oder tief
Kursiv fett
unterstreichen
NLQ-Schrift
in guter Qualität:
skgjöäüöÄüß.

Bild 3. Schriften wie beim FX-85 — inklusive Schönschrift

Ist der erste Eindruck gut, so wird dies durch unseren Drucktest (siehe Tabelle) bestätigt. Mit 1:50 Minuten für den Probetext liegt der Riteman II nur etwa sieben Sekunden hinter dem FX-85. Allerdings ergibt sich im NLQ-Modus mit 24 Zeichen/s ein größerer Abstand zum Vorbild (37 Zeichen/s). Auch sonst gibt es kleine Unterschiede. So ist beispielsweise die Druckmatrix im NLQ-Modus nur 16 x 12 Punkte groß. Allerdings ist das damit erreichte Druckbild sehr gut (Bild 2 und 3) und trägt den Namen Schönschrift zu Recht.

ESC — Eine Vielfalt der Steuermöglichkeiten

Alle Druckerfunktionen, teilweise auch die an den DIP-Schaltern einstellbaren, sind über ESC-Befehle anzusteuern. Dabei fällt auf, daß alle diese Befehle exakt mit denen des Epson FX-85 übereinstimmen. Nicht vorhanden ist jedoch der IBM-Modus (dafür gibt es eine eigene Riteman II-Version für 1298 Mark), der den FX-85 doch noch universeller sein läßt.

Der Riteman II ist mit seinem Preis von 1398 Mark ein Drucker, der sich hinter seinem Vorbild nicht zu verstecken braucht. Die Befehle des Riteman II orientieren sich stark am FX-85: Die Syntax ist meist die gleiche. Man kann den Riteman II als gelungene Synthese aus dem Leistungsangebot des FX-85 und dem Gedanken an ein kompaktes Gerät bezeichnen. Wer länger mit ihm arbeitet, wird sich bald über den zusätzlichen Platz auf dem Schreibtisch freuen, ohne auf wesentliche Funktionen verzichten zu müssen. Urteil: sehr empfehlenswert.

(Elisabeth Konther/aw)

MSP-10 — ein Uhrmacher auf Abwegen

Mit zwei Jahren Garantie auf die neu entwickelten Matrixdrucker will Citizen die Nummer 1 auf dem Druckermarkt werden.

Als Jack Bennet, Vizepräsident von Citizen Deutschland, am 26. September 1985 die neuen MSP-Matrixdrucker vorstellte, nannte er große Ziele. Nach seiner Aussage strebe man an, der weltweit führende Hersteller von qualitativ hochwertigen und extrem zuverlässigen EDV-Druckern zu werden. Der von uns getestete MSP-10 (Bild 1) ist mit 1648 Mark das bislang preiswerteste Instrument von Citizen zum Erreichen dieses Zieles — bleibt zu prüfen, ob es dazu geeignet ist.

Als ein Drucker mit Epson- und IBM-Modus wird der MSP-10 im deutschen Handbuch vorgestellt. Das Handbuch trägt seinen Namen zu Recht. Gut gegliedert und klar in der Darstellung gibt es auf (fast) alle Fragen Antwort. In diesem Handbuch stolpert man schon auf den ersten Seiten über eine Überschrift, die aufmerken läßt »Maintenance Self Test«. Dies läßt sich wohl am besten mit Wartungstestroutine übersetzen. Dieser Selbsttest gibt Auskunft darüber, was sich im Innern des Druckers abspielt. Als erstes zeigt diese Routine die eingebaute Version des Betriebssystems an, einschließlich des dazugehörigen Datums. Dann folgt in übersichtlicher, das heißt in lagerichtiger Ansicht, die Darstellung der Zustände der eingebauten zwei DIP-Schalter.

Die DIP-Schalter befinden sich leider im Innern des Druckers und sind von außen nicht zugänglich. Also heißt es, Papier heraus und Werkzeug her, sollte etwas geändert werden müssen. Nachdem die Unzugänglichkeit solcher Schalter immer wieder in den verschiedenen Tests bemängelt wird, ist es unverständlich, warum solche »Kleinigkeiten« noch falsch konstruiert werden. Da wir gerade bei den negativen Punkten sind, sei der zweite (und letzte) genannt. Der gesamte Drucker macht zwar einen optisch ausgezeichneten Eindruck, der sich aber beim ersten Berührungskontakt sofort verflüchtigt. Die Plastikabdeckung ist nur aufgelegt und auch der Papierseparator scheint es bei jeder Gelegenheit darauf anzulegen, aus der Halterung zu rutschen. Am besten ist es, wenn man beides gar nicht erst auspackt und den Drucker offen betreibt. Leider sind auch die Stachelwalzen beim MSP-10 so einfach auf der zugehörigen Achse befestigt, daß sie nur ungenau einzustellen sind. Selbst wenn die Walzen arretiert sind, kann man sie beinahe mühelos hin- und herschieben. Dieser Punkt bedarf einer dringenden Verbesserung, damit das sonst sehr gute Gesamtbild des MSP-10 nicht zu sehr getrübt wird. Doch zurück zum Handbuch. Daß ein (üblicher) Selbsttest vorhanden ist, muß nicht näher beschrieben werden. Anders dagegen der Hex-Dump-Modus des MSP-10. Durch Drücken von LF und FF während des Einschaltens angewählt, wird man beim Ausdruck, beispielsweise einer Textzeile, überrascht seinen Augen nicht trauen. Nicht nur die Hex-Werte erscheinen auf dem Papier, sondern auf der rechten Seite des Blattes auch die dazu gehörenden ASCII-Zeichen. Jeder, der schon einmal mit Handbüchern Hex-Werte in ASCII-Zeichen umgesetzt hat, wird diesen Komfort des MSP-10 schnell schätzen lernen.

Mit den Tasten ON LINE, LF und FF werden Hex-Dump, der Selbsttest, der Maintenance-Test und die Umschaltung in den NLQ-Modus vorgenommen. Das Einschalten des NLQ-Modus kann zeilenweise geschehen, das heißt, wird während eines Druckvorganges in den NLQ-Modus geschaltet, so wird die laufende Zeile ordnungsgemäß beendet und dann werden die folgenden Zeilen in Schönschrift (Bild 2 und 3) auf das Papier gebracht.

Auf dem »Mäuseklavier«, eine unter Technikern gebräuchliche Bezeichnung für DIP-Schalter, lassen sich unter anderem nicht weniger

Kurz belichtet

Tabelle. Umfassend — die Leistungen des MSP-10

Name des Druckers	Citizen MSP-10	empfohlener Preis	1648 Mark
Unterstreichen	Ja	Proportionalschr.	Ja
Zeichenmatrix	9 x 9	NLQ-Matrix	17 x 17
Papierbreiten	76,2 bis 254	Zeichenvorrat	ASCII 11 Intern
Papierarten	Einzel/Endlos	Durchschläge	2
Zeichen pro Zeile	0 bis 135	Selbsttest	Ja
Hexdump	Ja	Autom. Einzelbl.	Nein
Pufferspeicher	2 KByte	Rückwärtstransp.	Ja
Ladb. Zeichensatz	Ja	Probetext	1:46 Minuten
Geschwindigkeit	160,Messung 142	NLQ-Geschw.	29 Zeichen/s
Grafikmodi	bis 4fache Dichte (1920 Druckpositionen)		
Funktionstasten	ON Line, LF, FF, 16 Dip-Schalter		
Ausstattung	Handbuch deutsch, englisch, Farbband, Papierführ.		
Schriftarten	Pica, Elite, Schmal, Breit, Doppel, Fett, Italic		
Sonderfunktionen	Hex-Dump mit Hex- und ASCII-Darstellung		

Citizen MSP-10
Near Letter Quality
Normalschrift
Breit
Fettschrift
komprimiert
Unterstreichen
hoch- und tiefgestellt
Eliteschrift
Internationale Zeichen:ñÑ)¦¥åæÅ£

Bild 3. Eine kleine, aber feine Schriftprobe

als 11 internationale Zeichensätze auswählen. Außerdem sind dort, analog zum Epson, Papierlänge, die Form der Null, LF bei CR (Zeilenvorschub bei Wagenrücklauf), Papierendeerkennung und vieles mehr festzulegen. Allerdings nicht zu fest; denn alle diese Befehle können auch per Software (ESC) beliebig geändert werden. Da eine Änderung der Schalter die Demontage des MSP-10 erfordert, wollen solche Maßnahmen gut überlegt sein. Papierbreiten von 76,2 mm bis hin zu 254 mm sind keine besondere Aufgabe für den MSP-10. Dabei sind außer dem Original noch zwei Durchschläge möglich. Die Anschlagstärke wird an einem kleinen Hebel, seitlich am Gehäuse, verändert. Keine Frage, auch Einzelblätter verarbeitet der MSP-10 mühelos. Wegen der eingangs zitierten mechanischen Probleme sogar besser als Endlospapier.

Die hinlänglich bekannten, weil oft verwendeten ESC-Befehle dienen dem MSP-10 zur Steuerung von Format-Anweisungen, Schriftartenumschaltung und der Ausführung sonstiger Steuerbefehle (Tabelle).

Befehlsvielfalt

Im Vergleich mit unserem Referenzdrucker, dem FX-85, besitzt der MSP-10 ein stark erweitertes Angebot an ESC-Befehlen.

Mit Hilfe eines einzigen Befehls lassen sich unter anderem auch häufig benutzte Kombinationen aus Druckbild und Schriftart einstellen. Beispiele hierfür sind Pica/Schmalschrift, Elite/Normal, Elite/Schmal oder reverse Darstellung. Das tollste aber ist der als »Master Pitch« bezeichnete ESC-Befehl ESC(!) CHR$(n). Für die Variable n wird eine Zahl von 0 bis 255 eingesetzt. Diese Zahl ergibt sich als Summe einer Reihe von Wertigkeiten, die verschiedenen Druckbefehlen fest zugeordnet sind. Was sehr kompliziert klingt, ist in Wirklichkeit mit ein bißchen Routine sehr einfach anzuwenden. Ein Beispiel soll das Geschehen verdeutlichen: Pica(0) mit Doppeldruck(16), in Breitschrift(32) soll unterstrichen (128) werden. Diese Befehlskombination wird mit ESC(!) CHR$(176) (also mit der Summe der einzelnen Befehle) eingeleitet. Um den Rahmen dieses Artikels nicht zu sprengen, soll nur noch ein weiterer ungewöhnlicher Steuerbefehl erläutert werden. Mit »Backspace« wird der Befehl CHR$(8) bezeichnet. Damit bewirkt man das wiederholte Drucken, wobei der Druckkopf um eine Nadelbreite nach links versetzt wird. Beispielsweise kombiniert mit Kursivschrift und dem mehrfachen Wiederholen des Befehls ESC$(8) lassen sich völlig neue, ungewohnte Schriftbilder entwickeln. Der Kreativität sind (fast) keine Grenzen gesetzt.

Das Schriftbild des MSP-10 ist in allen Modi gleichmäßig und braucht den Vergleich mit anderen Druckern nicht zu scheuen. Auch die NLQ-Schrift läßt keine Wünsche mehr offen.



Bild 2. Die Schrift des MSP-10 kann sich auch vergrößert sehen lassen

Elektronik kontra Mechanik

Der MSP-10 ist ein Drucker, dessen elektronische Leistungsfähigkeit im Widerspruch zum mechanischen Aufbau des Gehäuses steht. Allerdings scheint man bei Citizen trotzdem größtes Vertrauen in die Mechanik zu haben, denn es wird eine bislang nie dagewesene Garantiezeit von zwei Jahren eingeräumt. Das selbstgesteckte Ziel wird Citizen mit dem MSP-10 aber wahrscheinlich nicht erreichen, denn dazu ist der Preis von 1648 Mark, in Anbetracht des nicht befriedigenden mechanischen Aufbaus, zu hoch. Bleibt zu hoffen, daß Citizen in Zukunft auch an seine Drucker die gleichen mechanischen Präzisionsanforderungen wie an seine Uhren stellt. (Elisabeth Konther/aw)

Info: Sylenec Datensysteme GmbH, Postfach 151727, 8000 München 15, Tel. (089) 5179-0

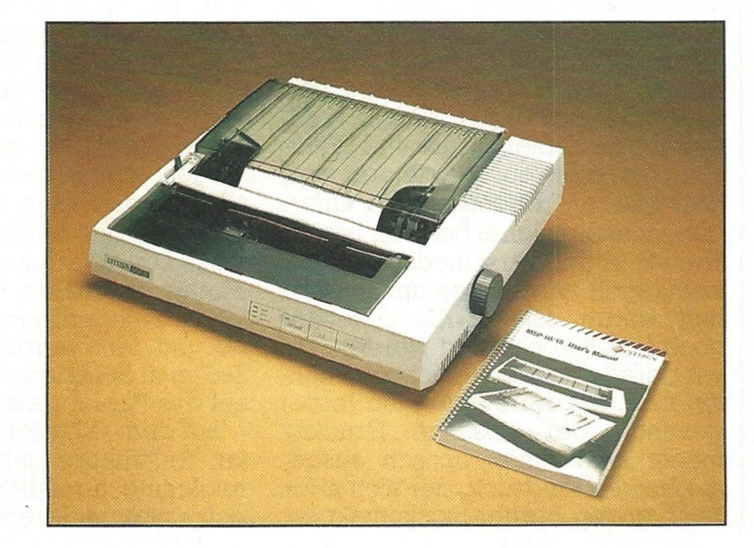


Bild 1. Das moderne Design des MSP-10 von Citizen



Großer Monitortest

Ein Monitor ist, zumindest bei Heimcomputern, das wichtigste Datenausgabegerät überhaupt. Wir stellen Ihnen eine Auswahl von 15 Monitoren vor, die für Heimcomputer wie den C 64 oder den C 128 in Frage kommen.

Wollen Sie Ihren Computer vom heimischen Fernseher trennen und an einen Monitor anschließen? Dann liegt Ihrem Entschluß wahrscheinlich einer der beiden folgenden Gründe zugrunde.

Der erste Grund: Sie werden ständig von Ihrer Familie bei der Programmwahl unterdrückt. »Schwarzwald-Klinik« statt 25 Zeilen mal 40 Zeichen.

Der zweite Grund: Die Qualität des Fernsehers reicht Ihnen nicht mehr aus. Das Bild ist Ihnen zu unscharf und die Fernseherröhre eventuell zu groß. Damit ist das Stichwort auch schon gefallen: Bildqualität. Da ein Monitor eine höhere Auflösung als ein Fernseher hat, ist ein schärferes und kontrastreicheres Bild möglich. Daß ein deutliches, kontrastreiches Bild die Augen weniger belastet als ein flaues, ist wohl fast selbstverständlich. Spätestens hier müssen die Monitore in zwei Klassen getrennt werden, den monochromen (schwarzweiß) und den Farb-Monitoren, da die monochromen Typen ein wesentlich deutlicheres Schriftbild haben als Color-Versionen. Ein monochromer Monitor für etwa 300 bis 400 Mark kann leicht 80 Zeichen pro Zeile lesbar auf den Bildschirm darstellen. Ein Farbmonitor mit FBAS- oder Composite-Eingang aber nur 40 Zeichen. 80 farbige Zeichen pro Zeile sind nur bei RGB-Monitoren möglich.

Sie sollten genau abwägen, ob Sie sich einen monochromen Monitor oder einen Farbmonitor zulegen. Verwenden Sie Ihren Computer als Schreibmaschine oder programmieren Sie beispielsweise naturwissenschaftliche Anwendungen, bei denen es auf Farbe nicht ankommt, dann sollten Sie sich einen monochromen Monitor kaufen. Sie tun Ihren Augen damit einen großen Gefallen. Sind Sie allerdings ein Spiele- oder Grafik-Fan, dann ist ein Farbmonitor für Sie das richtige Gerät. Beachten Sie auch, ob Sie einen Lautsprecher benötigen oder nicht. Die Videobandbreite eines Monitors gibt die Leistungsfähigkeit des Videoverstärkers an. Die Videobandbreite eines Monitors kann man mit dem Frequenzumfang eines Audioverstärkers vergleichen.

Sehr wichtig ist auch die Bildschirmdiagonale des Monitors. Für einen Heinmcomputer kommen 9 bis 14 Zoll in Frage, wobei sich 12 Zoll häufig als das Optimum erweisen. Allgemein: Je näher Sie den Monitor vor Augen haben, desto kleiner sollte der Bildschirm sein. 5- oder 6-Zoll-Monitore sind für die tägliche Arbeit zu klein.

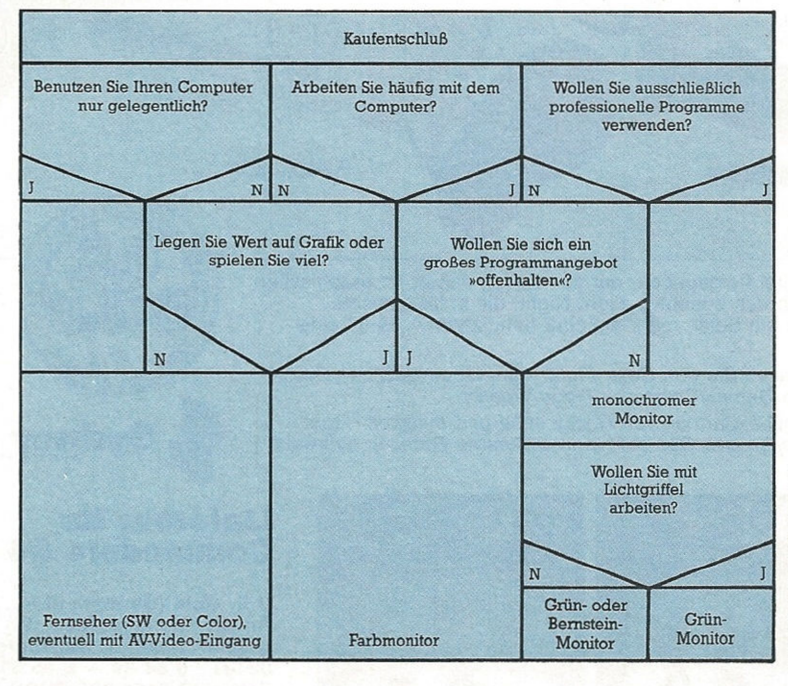
Wir haben 15 Monitore getestet. Acht Farbmonitore und sieben monochrome. Das Testbild wurde mit einem C 64 erzeugt, der 80-Zeichen-Text mit dem Textverarbeitungsprogramm Protext und dem C 128 im RGB-Modus. Zur Monitoransteuerung wurde das Luminanzsignal verwendet (siehe Monitorumschaltung in Ausgabe 10/85, Seite 16). Für das Testbild wurde absichtlich der C 64 verwandt und nicht der dazu kompatible C 128, da die Bildschirmdarstellung des C 128 im 40-Zeichen-Modus schlechter ist. Das Bild besitzt Längsstreifen.

Drei der getesteten Farbmonitore lassen sich an den C 128 sowohl im 80- als auch im 40-Zeichenmodus anschließen.

Zum Test der Monitore entwarfen wir ein Testbild, das die Farbqualität und die Auflösung eines Monitors testet. Da die hier getesteten Monitore wahrscheinlich nur an Heimcomputer angeschlossen werden, die kein »Superbild« liefern (können), haben wir auf einen Test mit einem Farbbildgenerator verzichtet. Denn hier würden eventuell Mängel sichtbar, die bei dem Bildsignal eines Heimcomputers »nie und nimmer« relevant werden.

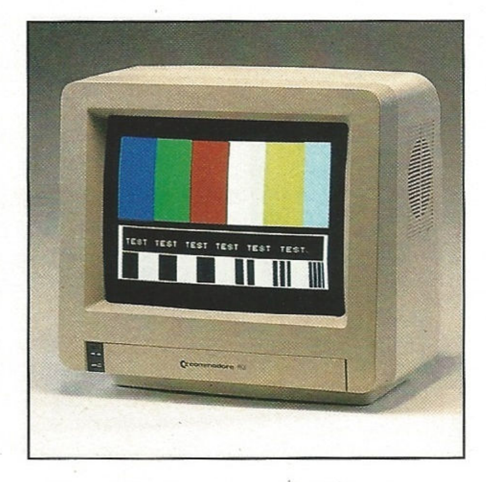
Unser Testbild zeigt, wie gut ein Monitor Farbwechsel und Schwarz/Weiß-Wechsel verkraften kann. Die vergrößerte Textdarstellung soll die Schärfe der Zeichendarstellung belegen. Im 80-Zeichenmodus haben wir bei der Schriftprobe verschiedene Zeichenfarben gewählt, um die Kontrastfähigkeit der RGB-Ansteuerung zu unterlegen.

Haben Sie Ihre Wahl getroffen, lassen Sie sich auf jeden Fall vom Händler ein Anschlußkabel für den Monitor an Ihren Computer geben. Testen Sie die Funktionsfähigkeit des Monitors und Anschlußkabels gleich im Laden. Denn teilweise sind Monitore schlecht abgeglichen: Das Bild ist verzogen, horizontal oder vertikal verschoben oder in der Mitte heller als am Rand. In einem solchen Fall sollten Sie den Monitor vom Händler vor einem Kauf exakt einstellen lassen oder auf einem anderen Gerät bestehen. (hm)



Ein »Nassi-Shneiderman-Diagram« zur Entscheidungsfindung beim Monitorkauf

Commodore 1901



Der Monitor zum C 128. Er »versteht« sowohl Composite- und RGB-Signale. Mit einem Schiebeschalter an der Frontseite wird zwischen beiden Normen umgeschaltet.

Die Bildqualtität des 1901 ist als sehr gut einzustufen gemessen an den Mitbewerbern. Schwarz/Weiß-Wechsel werden sehr gut verkraftet. Bei einem Wechsel der Grundfarben treten zwischen den Farbbalken schwarze Linien auf. Beim 1901 hält sich dieses Phänomen, das bei jedem getesteten Farbmonitor auftrat, aber noch in Grenzen. Unschön sind die »Geisterbilder«. Steht am Zeilenanfang ein helles Zeichen, wird die ganze nachfolgende Zeile heller. Durch eine geeignete Kontrast- und Helligkeitseinstellung kann dieses Phänomen aber soweit unterdrückt werden, daß es kaum noch ins Gewicht fällt.

Der RGB-Modus besticht durch eine exzellente Schärfe der Zeichen, obwohl der 1901 »nur« eine Lochmaske besitzt. Das zeigt, was ein guter Konvergenzabgleich ausmacht.

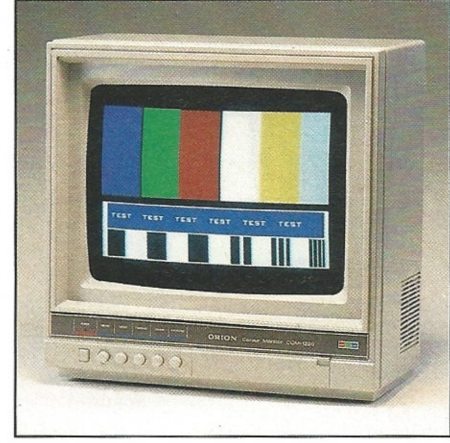
Positiv
Bedienungselement an der Front
Eingebautes Audioteil

Negativ
Kein mitgeliefertes Kabel
»Geisterbilder«



Info: Commodore Büromaschinen, Lyoner Str. 38, 6000 Frankfurt, 069/6638-0, Preis: 1098 Mark

Orion CCM 1280



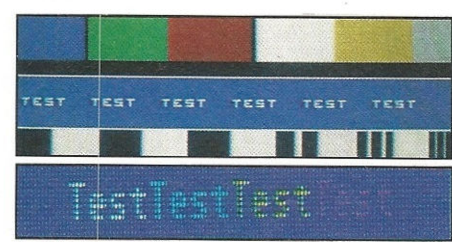
Auch dieser Monitor läßt sich ohne weiteres an den C 128 anschließen. Dazu wird ein Kabel mitgeliefert. Zwischen dem Composite- und dem RGB-Modus wird mit einem Schalter gewählt. Leider ist dieser Schalter an der Rückseite nur schwer zu erreichen.

Der Monitor kann auch zwischen RGB-TTL und RGB-Analog umgeschaltet werden. Beim Composite- und FBAS-Eingang heißt es aufpassen: »Separate« bedeutet Composite und »Composite« bedeutet FBAS. Außerdem sind die Farben der Chrominanz- und Luminanz-Buchse vertauscht. Die Zeichenschärfe im 80-Zeichenmodus ist subjektiv etwas geringer als die des 1901, trotz der Schlitzmaske.

Im Composite-Modus kann ein höherer Kontrast als beim 1901 eingestellt werden. Bei Schwarz/Weiß-Wechseln sind leichte Unschärfen an den Kanten festzustellen, die aber noch vernachlässigt werden können. Beim Farbbalken-Test ist der 1901 dem Orion überlegen.

Positiv
Eingebautes Audioteil
FBAS-Eingang vorhanden

Negativ
RGB/Composite-Umschalter an der Rückseite
Benennung der Eingänge verwirrend



Info: Bezugsquelle: Hard&Soft, Gagernstr. 4, 8580 Bayreuth, 0921/68877, Preis: 998 Mark

Picom 4B/PR



Der Picom hat einen FBAS- und einen RGB-TTL/Analog-Eingang. Wir haben den Picom 4B/PR an den C 128 angeschlossen. Die Bildqualität der 80-Zeichendarstellung ist ähnlich der des Orion. Zum Anschluß an den C 128 wird das gleiche Kabel wie beim Orion benötigt: 9pol. Cannon — 8pol. VTR. Doch nun zum FBAS-Modus des Picom 4B/PR RGB/PAL.

Das Testbild zeigt, daß der Picom Farbwechsel gut verkraftet. Nur zwischen Grün und Rot ist eine schwarze Kante vorhanden. In dieser Disziplin schlägt der Picom (Schlitzmaske) den 1702. Beim Linientest zeigt der Picom allerdings Schwächen: Weiße Linien auf schwarzem Grund werden farbig.

In der Textschärfe beim Test mit 40 Zeichen pro Zeile ist der 4B/PR dem 1702 unterlegen.

Die an der Rückseite angebrachten Regler zur Kontrast- und Farbeinstellung müssen mit einem Schraubendreher eingestellt werden. Das ist eine wesentliche Einschränkung des Bedienkomforts.

Positiv
Gutes Ergebnis im Farbbalkentest

Negativ
Kein Audioteil
Regler an der Rückseite, nur mit Schraubendreher zugänglich



Info: Mirwald Elektronik, Fasanenstr. 8, 8025 Unterhaching, 089/6111224, Preis: 1548 Mark

Commodore 1702



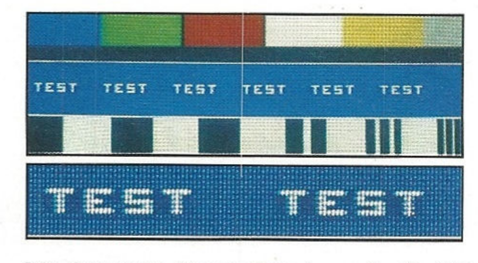
Ein guter alter Bekannter, der von Commodore als Monitor für den C 64 angeboten wird. Der 1702 besitzt ein sehr gutes Bild (Schlitzmaske).

Bei Schwarz/Weiß-Wechseln treten so gut wie keine Farbverschiebungen oder Unschärfen auf. Sehr gut ist das an den Linien im Testbild zu sehen. Selbst die Farbwechsel Blau-Rot-Grün werden ohne größeres »Ausreißen« der Farben wiedergegeben.

Beim 1702 sind alle Bedienungselemente hinter einer Klappe an der Frontseite versteckt. An der Rückseite befindet sich nur der Umschalter für Composite und FBAS, der allerdings selten benötigt wird. Zum Beispiel dann, wenn Sie einen Videorecorder oder einen VC 20 an den 1702 anschließen möchten. Das FBAS-Signal kann über die Buchse an der Frontseite eingespeist werden. Die Audiobuchse an der Frontseite ist mit der an der Rückseite parallel geschaltet. Der Umschalter schaltet lediglich die Videobuchsen um. Beim Anschluß zweier Tonquellen führt das zu Problemen.

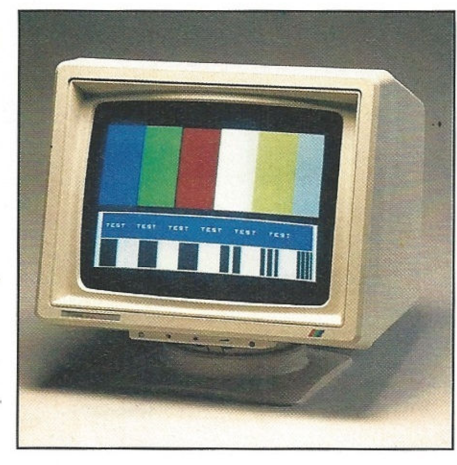
Positiv
Sehr guter Kontrast
Eingebautes Audioteil

Negativ
Parallelschaltung der Audiobuchsen an der Front- und Rückseite.



Info: Commodore Büromaschinen, Lyoner Str. 38, 6000 Frankfurt, 069/6638-0, Preis: etwa 750 Mark

Cable MC3700



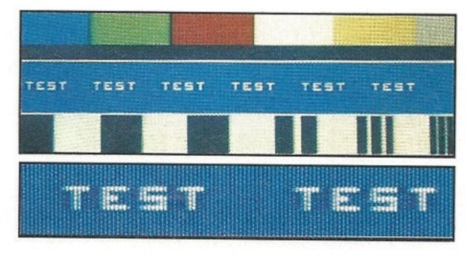
Von der Bildqualität — Kontrast und Schärfe — ist der MC3700 mit dem 1702 vergleichbar. Auch beim Farbbalken- und Linientest sind zwischen dem 1702 und dem Cable kaum Unterschiede festzustellen. Die Bildröhre des MC3700 hat eine Schlitzmaske. Durch den im Lieferumfang enthaltenen Schwenkfuß kann der MC3700 optimal auf die Arbeitsposition ausgerichtet werden.

Sehr negativ auf die Bedienungsfreundlichkeit wirkt sich der an der Rückseite angebrachte Ein-/Ausschalter aus. Die Röhre des MC3700 wird per Hand entmagnetisiert. Ein Vorgang, der der Vergangenheit angehört. Ein heutiger Monitor sollte das automatisch machen.

Die Video- und Tonsignale werden dem Cable über eine 6polige Buchse zugeführt. Mit einem Schiebeschalter kann man wählen, ob ein FBAS- oder Chrominanz-Signal verwendet werden soll. Die Ausschnittvergrößerung zeigt die Zeichenschärfe bei der Farbkombination weiß auf dunkelblau.

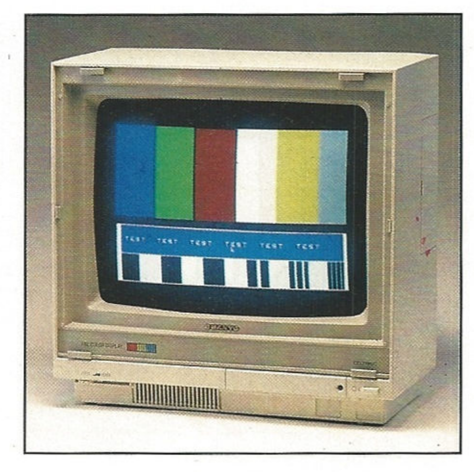
Positiv
Sehr guter Kontrast
Schwenkfuß
Audioteil

Negativ
Netzschalter an der Rückseite
Manuelles Entmagnetisieren



Info: Bezugsquelle: Boston Computer, Rosenheimer Str. 145a, 8000 München 80, 089/491073, Preis: 798 Mark

Sanyo CD3195C



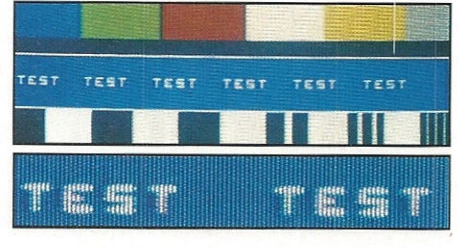
Der Sanyo CD3195C verfügt über eine befriedigende Bildqualität. Beim Test mit Schwarz/Weiß-Wechseln ist der 1702 dem Sanyo überlegen. Ebenso in der Schärfe der Schriftwiedergabe. Beim Farbbalken-Test sind gegenüber dem 1702 keine Unterschiede festzustellen.

Der Netzschalter und Lautstärkeregler befinden sich auf der Gerätefrontseite. Alle anderen Regler für Kontrast, Farbsättigung und Helligkeit etc. befinden sich hinter einer Klappe an der rechten Gehäuseseite. Hinter dieser Klappe verbirgt sich auch der Umschalter für Farb- und Gründarstellung. Durch diese Grünumschaltung wird das Arbeiten mit Texten, zum Beispiel bei Textverarbeitungsprogrammen oder beim Programmieren erleichtert. Wegen dieser Klappe muß der Sanyo CD3195C auf der rechten Seite freistehen, um »kurz mal« die Einstellung ändern zu können.

Der Sanyo kann mit FBAS- und Composite-Signalen angesteuert werden. Die Normumschaltung erfolgt an der Rückseite.

Positiv
Farb/Grünumschaltung
Audioteil

Negativ
Regler an der Seite hinter Klappe



Info: Sanyo, Lange Reihe 29, 2000 Hamburg 1, 040/2801045, Preis: 798 Mark

Taxan Vision Ex



Der Vision Ex hat neben einem FBAS- auch einen RGB-Eingang. Allerdings konnte der Vision Ex nicht am C 128 betrieben werden. Grund: die Pegel der Sync-Signale. Wir haben die Sync-Signale invertiert und der Vision Ex funktionierte. Aus diesem Grund haben wir den Vision Ex nur über den FBAS-Eingang getestet.

Wie Sie im Testbild sehen können. besteht der Vision Ex den Farbbalken-Test, wie der Picom, mit Bravour. Etwas schwer tut er sich im Linientest. Die eigentlich weißen Linien werden von Farbkanten gesäumt.

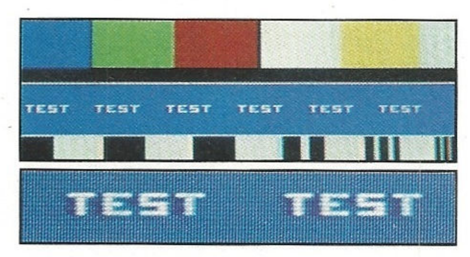
Die Zeichenschärfe ist nur geringfügig schlechter als die des 1702, was aber kaum auffällt, da der Vision Ex einen kleineren Bildschirm hat (12 Zoll).

Leider hat der Vision Ex nur einen FBAS-Eingang. Ein getrennter Composite-Eingang hätte die Leistung wahrscheinlich deutlich verbessert.

An der Frontseite befinden sich beim Vision Ex nur Netzschalter, Kontrast- und der Lautstärkeregler.

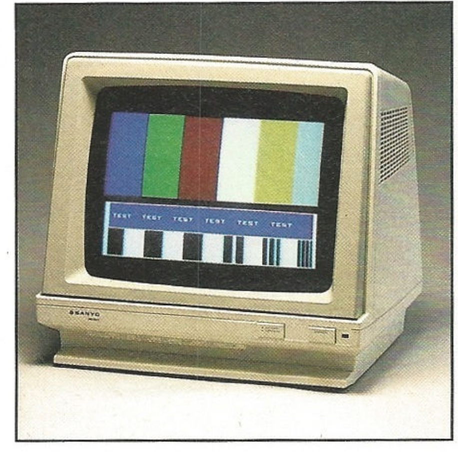
Positiv
Gutes Ergebnis im Farbbalkentest
Eingebautes Audioteil

Negativ
Fast alle Einstellungen erfolgen an der Rückseite.



Info: Melchers & Co, Schlachte 39/40, 2800 Bremen 1, 04211/176989, Preis: 1198 Mark

Sanyo DMC 6655



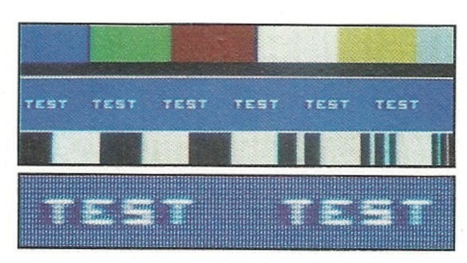
Ebenfalls ein Monitor mit zusätzlichem RGB-Eingang (SCART-Buchse). Der Anschluß des C 128 ist an diesen Monitor nicht ohne weiteres möglich. Die RGB-Signale des C 128 müssen zum Anschluß über Spannungsteiler abgeschwächt werden, damit der Monitor nicht übersteuert wird. Hat man die Anpassung erledigt, sind die 80 Zeichen pro Zeile des C 128 durchaus lesbar.

Wir haben deshalb den DMC 6655 nur mit einem C 64 getestet. Im Farbbalken-Test ist der Sanyo DMC 6655 etwas besser als der 1702. Nicht aber im Linientest. Weiße Linien auf schwarzem Grund werden farbig wiedergegeben. Die Schärfe der Schrift kann als gut bezeichnet werden.

Wie der Taxan Vision Ex, wird der DMC 6655 mit einem FBAS-Signal angesteuert, das einfach nicht die Bildqualität übertragen kann als die Composite-Norm mit getrennten Luminanz- und Chrominanzsignal. Wie sein Bruder, der CD3195C, hat auch der DMC 6655 einen Grünschalter für bessere Textwiedergabe.

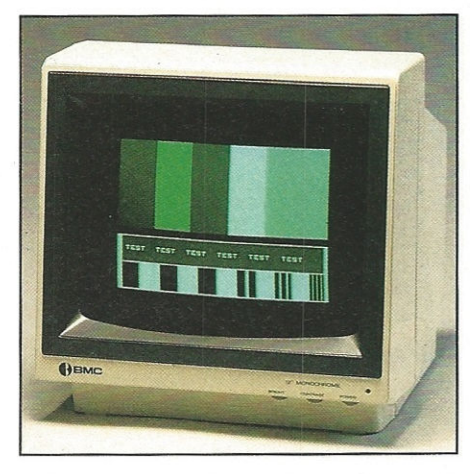
Positiv
Eingebautes Audioteil
Grünschalter
Alle Bedienelemente an der Frontseite

Negativ
Nur über Spannnungsteiler an C 128 anschließbar



Info: Sanyo, Lange Reihe 29, 2000 Hamburg 1, 040/2801045, Preis: 998 Mark

BMC BM-12G



Der BMC BM-12G konnte durch die beste 80-Zeichendarstellung überzeugen. Die Bildschirmfarbe ist, wie weithin üblich, grün. Für Grün ist das menschliche Auge am empfindlichsten. Das heißt, daß für das Auge Grün auf Schwarz einen größeren Kontrast bildet als Orange auf Schwarz. Viele Computerbesitzer geben deshalb Grünmonitoren den Vorzug.

Wie bei monochromen Monitoren üblich, wird der BMC-Monitor über ein Luminanz (BAS)-Signal angesteuert.

In die Eingangsleitung kann über einen Schiebeschalter an der Rückseite ein 75-Ohm-Abschlußwiderstand geschaltet werden. Die Videobuchse des BM-12G ist »durchgeschleift«. Dadurch könnte beispielsweise ein Videorecorder zu Dokumentationszwecken, neben dem Computer, an den Monitor angeschlossen werden. Die Anschlüsse erfolgen über Cinch-Buchsen.

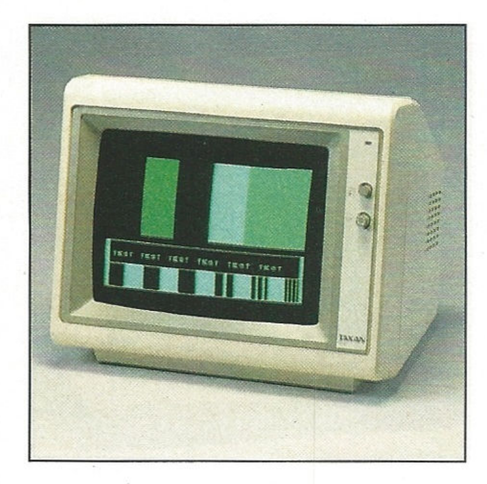
Die Regler für Kontrast und Helligkeit befinden sich an der Frontseite.

Positiv
Sehr gute Textwiedergabe
Kontrast- und Helligkeitsregler an der Frontseite.

Negativ
Kein Audioteil

Info: Mirwald Elektronik, Fasanenstr. 8, 8025 Unterhaching, 089/6111224, Preis: 379 Mark

Taxan KX1201E



Die Qualität der Textdarstellung ist von der des BMC-12G kaum zu unterscheiden. Die etwas höhere Schärfe des BMC-12G läßt sich nur durch genaues Hinschauen erkennen.

Wie der BMC-12G hat auch der Taxan KX1201 kein Audioteil. Also keinen Tonverstärker und keinen Lautsprecher. In dieser Ausgabe finden Sie eine Anleitung, wie Sie ein Tonteil selbst bauen können.

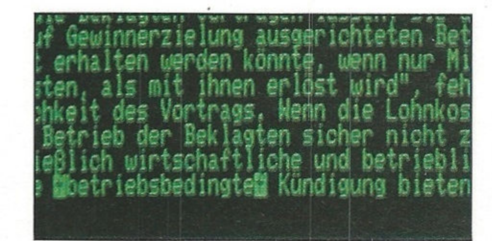
An der Frontseite befindet sich der Netzschalter und der Helligkeitsregler. Der Kontrast wird an der Rückseite eingestellt.

Angesteuert wird der Taxan mit dem Luminanzsignal des C 64 und nicht mit dem FBAS-Signal, das zu einer »streifigen« Darstellung führt. Das gilt für jeden monochromen Monitor, da die Farbanteile im FBAS-Signal hier nicht benötigt werden und die Helligkeitsinformation unnötig überlagern.

Der Taxan KX1201E läuft bei uns in der Redaktion seit mehr als einem Jahr ohne Störungen.

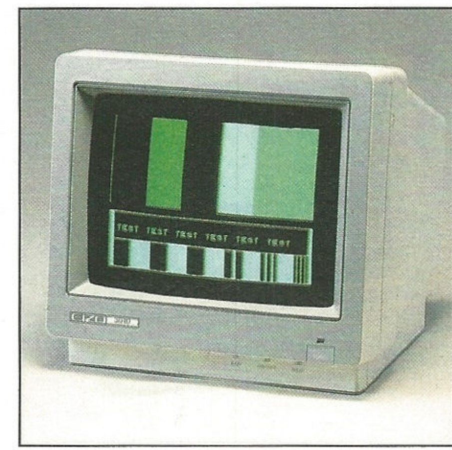
Positiv
Gute Bildwiedergabe

Negativ
Kontrastregler an der Rückseite
Kein Audioteil



Info: Melchers & Co, Schlachte 39/40, 2800 Bremen 1, 0421 1/176989, Preis: 398 Mark

Eizo 3010G



Ein Qualitätsunterschied des Bildes zum Taxan KX 1201E ist kaum festzustellen. Wenn Sie das Testbild betrachten, können Sie eigentlich nur die geringfügig andere Farbe der Leuchtschicht des Eizo feststellen, was aber auch auf eventuell unterschiedliche Kontrasteinstellungen zurückgeführt werden kann.

Beim Eizo 3010 sind alle Bedienungselemente, Netzschalter, Kontrast- und Helligkeitsregler an der Frontseite angebracht und somit gut erreichbar.

Der Eizo 3010G hat ein Audioteil, also einen Verstärker und Lautsprecher, eingebaut. Der Regler für die Lautstärke befindet sich ebenfalls an der Frontseite.

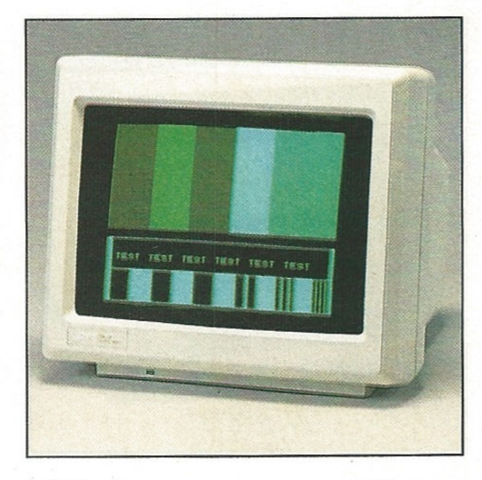
Das Audiosignal wird über eine Klinkenbuchse zugeführt. Das Luminanzsignal über eine Cinchbuchse. Die Empfindlichkeit des Luminanzeingangs reicht für den C 64 gerade noch aus. Die Bildröhren des Taxan und des BMC lassen sich deutlich heller einstellen.

Positiv
Alle Bedienelemente an der Frontseite
Eingebautes Audioteil

Negativ
Geringe Empfindlichkeit des Luminanzeingangs
Lange Einschaltzeit

Info: Rein Elektronik, Lötscher Weg 66, 4054 Nettetal 1, Preis: 492 Mark

Zenith CVM 1230



Wieder ein »stummer« Monitor. Der Zenith hat von den getesteten monochromen Monitoren das optisch ansprechendste und auch das kleinste Gehäuse, obwohl alle getesteten Monitore mit einer 12-Zoll-Bildröhre ausgestattet waren. Dem Zenith ist also der Vorzug einzuräumen, wenn Platzprobleme gelöst werden müssen.

Die Regler für Kontrast und Helligkeit befinden sich auf der rechten Gehäuseseite. Der Netzschalter ist ungünstigerweise an der Rückseite angebracht, was das Ein- und Ausschalten des Monitors schnell lästig werden läßt.

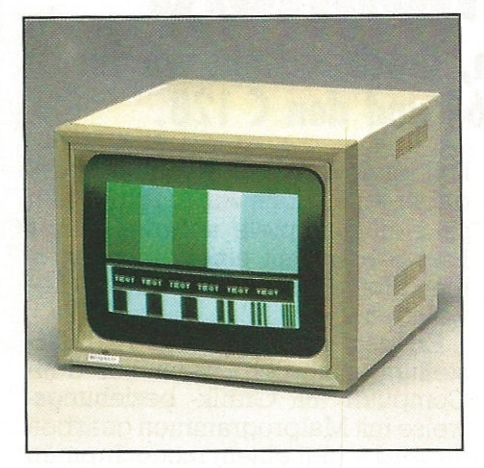
Bei der Bildqualität kann man keine wesentlichen Unterschiede gegenüber dem Eizo und dem Taxan feststellen. Kontrast und Helligkeit können in weiten Grenzen variiert werden. Das ist wichtig, denn eine Änderung der Farbeinstellung von Zeichen- und Hintergrundfarbe macht meist auch eine Änderung der Kontrast- und Helligkeitseinstellung notwendig.

Positiv
Kleine Gehäuseabmessungen
Ansprechendes Design

Negativ
Kein eingebautes Audioteil
Netzschalter an der Rückseite

Info: Zenith Datensysteme, Robert-Bosch-Str. 32-38, 6072 Dreieich-Sprendl, Preis: 395 Mark

Watanabe CD12 GR



Der CD12 präsentiert sich im professionellen Stahlblechgehäuse, wie man es von Geräten in der Industrie her kennt.

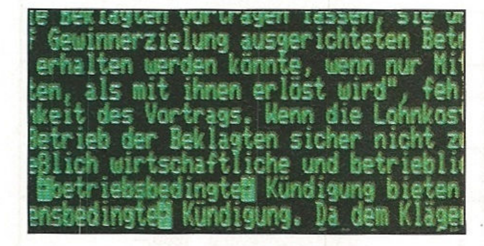
Alle Einstellelemente befinden sich an der Rückseite, auch der Netzschalter, ein kleiner Kippschalter. Der CD 12 hat kein Audioteil eingebaut.

Der Luminanzeingang ist durchgeschleift. Es können also zusätzliche Video(aufnahme)geräte an den Monitor angeschlossen werden und vom Computer das Signal erhalten. Der Luminanzeingang und -ausgang liegen an BNC-Buchsen an. BNC-Stecker und Buchsen erlauben sehr gute und mechanisch feste Hochfrequenz-Verbindungen. In die Luminanzleitung kann über einen Schiebeschalter ein 75-Ohm-Abschlußwiderstand geschaltet werden. Dieser Anschlußwiderstand ist dann erforderlich, wenn das Eingangssignal zu hoch ist, so daß das Bild »übersteuert« ist.

Die Bildqualität entspricht der des Eizo und Taxan.

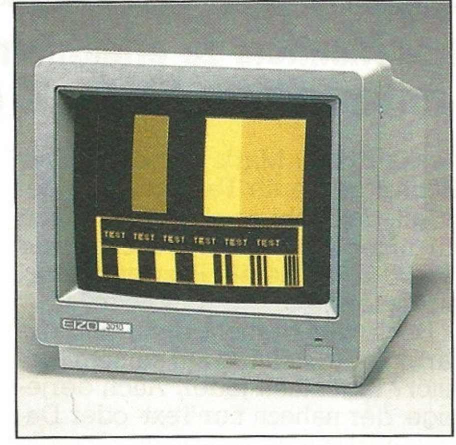
Positiv
Stahlblechgehäuse für Laboreinsatz

Negativ
Regler und Netzschalter an der Rückseite
Kein Audioteil



Info: Watanabe, Postfach 1155, 8036 Herrsching, Preis: 510 Mark

Eizo 3010A



Das gelbe Pendant zum 3010G. Von außen unterscheidet sich der Bernsteinmonitor nicht vom Grünmonitor.

Über bernsteinfarbige Bildschirme gehen die Meinungen auseinander. Viele Leute bevorzugen Grünmonitore wegen des höheren Kontrasts. Andere halten mehr von bernsteinfarbigen, da hier die Leuchtschicht eine wesentlich längere Nachleuchtdauer hat als bei Grünmonitoren und dadurch kein Flimmern mehr auftritt. Wegen der langen Nachleuchtdauer eignet sich ein Bernsteinmonitor nicht zur Ansteuerung eines Lichtgriffels. Man kann nicht mehr feststellen, wenn der Zeilenstrahl der Röhre den Fototransistor im Lichtgriffel durchschaltet.

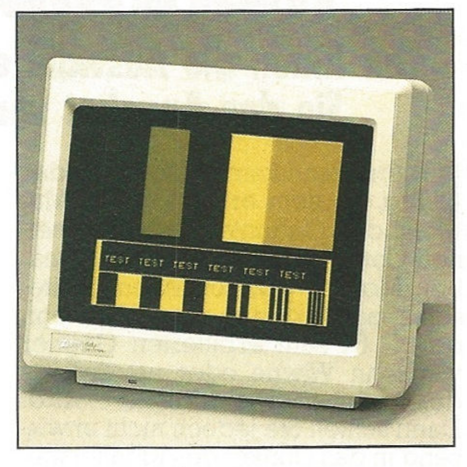
Wie der 3010G hat auch der 3010A einen recht unempfindlichen Luminanzeingang. Mit einem C 64 ist die Bildschirmdarstellung fast schon zu dunkel.

Positiv
Alle Bedienelemente an der Frontseite
Eingebautes Audioteil

Negativ
Geringe Empfindlichkeit des Luminanzeingangs
Lange Einschaltzeit

Info: Rein Elektronik, Lötscher Weg 66, 4054 Nettetal 1, Preis: 508 Mark

Zenith CVM 1220



Hier gilt das gleiche wie beim Eizo 3010G. Der CVM 1220 unterscheidet sich vom CVM 1230 nur in der Bildschirmfarbe. Bernstein anstelle von Grün. Von der Schärfe der Zeichen ist der CVM 1220 mit dem CVM 1230 vergleichbar. Auch hier ist der Netzschalter wieder an der Rückseite angebracht.

Wie schon beim Eizo 3010 erwähnt ist die Wahl »Bernstein oder Grün« eine Sache des persönlichen Farbempfindens. Man kann nicht sagen, ein grüner Monitor sei besser als ein bernsteinfarbiger oder umgekehrt. Wir raten Ihnen, vor einem Kauf sich einmal beide Typen im Laden vorführen zu lassen. Eventuell gefällt Ihnen weder ein Grün- noch ein Orangemonitor, sondern vielleicht ein Weißmonitor, bei dem die Schrift weiß ist. Wegen der Seltenheit der Weißmonitore haben wir kein solches Gerät getestet. Das gleiche gilt für Blau-Monitore, die der »letzte Schrei« sein sollen.

Positiv
Kleine Gehäuseabmessungen
Ansprechendes Design

Negativ
Kein eingebautes Audioteil
Netzschalter an der Rückseite

Info: Zenith Datensysteme, Robert-Bosch-Str. 32-38, 6072 Dreieich-Sprendl, Preis: 395 Mark

Marktübersicht Farbmonitore

Um Ihnen die Auswahl eines Monitors zu erleichtern, durchleuchten wir für Sie das Angebot von Farbmonitoren für den C 64 und den C 128.

Irgendwann kommt für jeden der Tag, an dem er zum ersten Mal mit den Gedanken spielt, an seinen Computer einen standesgemäßen Monitor anzuschließen. Spätestens passiert das nämlich, wenn der C 64 zum erbitterten Gegner von John Wayne oder neuerdings der »Schwarzwald-Klinik« wird. Dann sollten Sie jedoch nicht unwissend in das Dunkel des Monitorwaldes schreiten, sondern sich vorher unter Zuhilfenahme dieser Marktübersicht auf den Kauf vorbereiten.

Zunächst wollen wir einmal klären, was mit »standesgemäß« gemeint ist.

Erstens: Die Qualität der Darstellung ist bei einem Monitor besser als bei einem optimal eingestellten Fernseher. Wenn ein Computer ein Composite-, oder gar ein RGB-Signal zur Verfügung stellt, wie es beim C 64 beziehungsweise beim C 128 der Fall ist, dann sollte man das Signal auch nutzen. Die Bildqualität wird Sie überzeugen.

Zweitens: Ein Monitor muß nicht mehr kosten als Computer, Floppy und Drucker zusammen. Wir haben deshalb 1800 Mark als vertretbare Höchstgrenze für Farbmonitore gewählt, gute gibt es aber auch schon für unter 1000 Mark. Dabei wurde der C 128/C 64 als Basisgerät vorausgesetzt.

Doch wer braucht denn nun die farbige Darstellung auf seinem Monitor? Eigentlich jeder. Auch derjenige, der nahezu nur Text- oder Datenverarbeitung auf seinem Computer betreibt. Einige Programme setzen die Farbe ein, um zum Beispiel Fehler oder einen bestimmten Modus anzuzeigen. Vorsicht ist geboten, wenn 80 Zeichen dargestellt werden sollen (siehe Test ab Seite 24). Nicht alle Farbmonitore können 80 Zeichen so darstellen, daß ermüdungsfreies Arbeiten möglich ist, da sind monochrome Monitore besser und auch billiger.

Mehr als nur eine Frage des Preises

Auf einem Farbmonitor fällt das Erkennen von einzelnen Gegenständen, vor allem bei Spielen, wesentlich leichter. Die Gewohnheit spielt dabei eine große Rolle. Außerdem sind einige Spiele aufgrund Ihrer Farbgebung mit einem monochromen Monitor nur sehr schwer zu bewältigen.

Ebenso wichtig ist die Farbdarstellung natürlich, wenn auf dem Computer mit Grafik- beziehungsweise mit Malprogrammen gearbeitet wird — auf einem monochromen Monitor erhält man nicht einmal einen Teil des Gesamteindruckes einer Farbgrafik. Zusammenfassend ist zu sagen, daß, wenn man nicht allzu sehr auf den Preis schauen muß, ein Farbmonitor als wünschenswert angesehen werden kann. Nur wenn Sie auf Grafik und Farbe verzichten können, kann ein monochromer Monitor eventuell die bessere Wahl sein.

Um Ihnen eine Vorauswahl zu ermöglichen, haben wir die Marktübersicht (Tabelle) zusammengestellt. Alle Daten beruhen auf Herstellerangaben. Die Marktübersicht erhebt keinen Anspruch auf Vollständigkeit.

(og)

Marktübersicht Farbmonitore

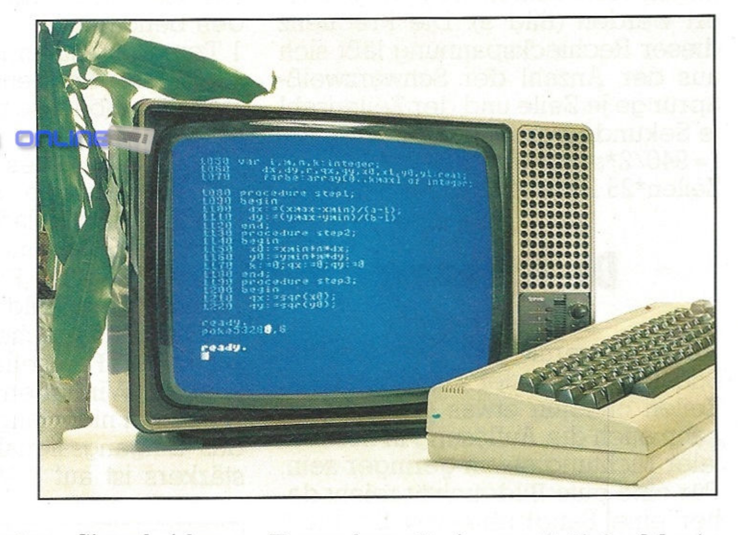
Anbieter	Typ	Signaleingänge L=Luminanz C=Chrominanz Rd=RGB digital Ra=RGB analog	Anschlußbuchsen C=Cinch 8=8pol. Kleinger.-Stecker 9=9pol. Cannon	Bilddiagonale in Zoll	Audioteil mit Lautsprecher j=ja n=nein	Entspiegelt	Alle Bedienungselemente an der Frontseite	Videokabel für den C 64 lieferbar? Sonst Anschluß an den C 128	Preis inkl. MwSt.
Ce-tec	FTC 1201 P/R	FBAS, Rd	C, 8	12	j	j	j	j, 69,—	1498,—
	FTC 1201 R	Rd	8	12	n	j	j	j, 69,—	1498,—
	FTC 1410 R	Rd	8	14	n	j	j	j, 69,—	1498,—
	FTC 1410 P/R	FBAS, Rd	C, 8	14	j	j	j	j, 69,—	1498,—
Commodore	1702	L, C, FBAS	C	14	j	n	j	j	ca. 750,—
	1902	L, C, FBAS, Rd	C, 9	13	j	j	j	j	ca. 998,—
Cosmos Powerline	RT 2000 F14	Rd, Ra	C, 10	14	n	j	—	n	k. A.
Feltron Zeissler	CM-421	Rd	9	14	n	j	j	n	1425,—
Gerb Computer	NOVEX-1414-CL	Rd	C, 8	14	j	n	j	j	798,—
Grundig	PM 015 RGB	FBAS, Ra	Scart	14	j	n	j	n	auf Anfrage
Hantarex	CT 900/3 SR 14	L, C	7	14	j	n	j	j, 30,—	998,—
	CT 900/1 SR 4	FBAS, Rd, Ra	6pol. DIN	14	j	n	j	j, 30,50	1125,—
Loewe Opta	DM 114	L, C, FBAS, Ra	Scart	14	j	n	j	j, 21,45	1254,—
MAGNA	MAGNA Modell C 64	L, C	7	14	j	n	j	j	1190,—
Mirwald Electronic	4B/PR	FBAS, Rd, Ra	8	14	n	n	j	j, 79,—	1584,60
	BM 1010	Rd	8	12	n	j	j	n	auf Anfrage
NEC	JC-1420 DE	Rd	8	14	n	j	j	n	1110,—
	JC-1210 DFE	Rd	8	12	n	j	j	n	1280,—
	JC-1203 DHE-5	Rd	8	12	n	j	j	n	1590,—

Anbieter	Typ	Signaleingänge L=Luminanz C=Chrominanz Rd=RGB digital Ra=RGB analog	Anschlußbuchsen C=Cinch 8=8pol. Kleinger.-Stecker 9=9pol. Cannon	Bilddiagonale in Zoll	Audioteil mit Lautsprecher j=ja n=nein	Entspiegelt	Alle Bedienungselemente an der Frontseite	Videokabel für den C 64 lieferbar? Sonst Anschluß an den C 128	Preis inkl. MwSt.
Räbiger	TM04	L, FBAS	5pol. DIN	14	j	n	n	j, 8,—	729,—
	TM01	L, C, Rd, Ra	5pol. DIN	14	n	n	n	—	698,—
Rein	EIZO 8010	L, C, FBAS, Rd, Ra	C, 8	14	j	n	j	j, 21,—	1170,—
	EIZO 7030 M	Rd, Ra	8	12	n	j	j	—	1351,—
	EIZO 7030 H	Rd, Ra	8	12	n	j	j	—	1505,—
Sanyo	CD 3220N	Rd	8	14	n	n	j	j, 69,—	899,—
	DMC 6650	RD	8	14	n	n	j	j, 69,—	998,—
	CD 3195 C	L, C, FBAS	C	14	j	n	n	j	798,—
Sony	CPD-1000	L, Rd, Ra	—	10	n	j	n	nur C 128	1698,—
	CPD-1301	L, Rd, Ra	—	13	n	j	n	nur C 128	1798,—
Taxan	Vision PAL	FBAS	C	14	j	n	j	j	898,—
	Vision Ex+	FBAS, Rd	C, 8	12	j	j	j	j	1198,—
	Vision III+	Rd	8	12	n	j	j	n	1799,—
	Vision-PC+	Rd	9	12	n	j	j	n	1799,—
Zenith	Z133e	Rd, Ra	C, 9	13	n	j	j	j, 80,—	1000,— bis 1500,—

Ce-tec Trading GmbH, Lange Reihe 29, D-2000 Hamburg 1; Commodore Büromaschinen GmbH, Lyonerstraße 38, 6000 Frankfurt/Main 71; Cosmos Powerline GmbH, Winzererstraße 47D, 8000 München 40; Feltron Zeissler & Co. GmbH, Auf dem Schellerod 22, D-5210 Troisdorf; Gerb Computer GmbH, Roedernallee 174-176, 1000 Berlin 51; Grundig Electronic, Würzburger Straße 150, 8510 Fürth/Bay.; Hantarex - Deutschland, Siegner Straße 23, 5230 Altenkirchen; Loewe Opta GmbH, Industriestraße 11, 8640 Kronach; Magna Vertriebsges. mbH, Hauptstraße 1, 6384 Schmitten 2; Mirwald Electronic, Fasanenstraße 8, 8025 Unterhaching; NEC Home Electronics (Europe) GmbH, Wiesenstraße 148, 4040 Neuss 1; Peter Räbiger Microcomputer-Systeme, Veldener Straße 65, 5160 Düren; Rein Elektronik GmbH, Lötscher Weg 66, 4054 Nettetal 1; Sanyo Video Vertrieb GmbH & Co., Lange Reihe 29, 2000 Hamburg 1; Sony Deutschland GmbH, Hugo-Eckener-Straße 20, 5000 Köln 30; Taxan Vertriebsgesellschaft, Schlachte 39/40, 2800 Bremen; Zenith Data Systems, Robert-Bosch-Str. 32—36, 6072 Dreieich-Sprendl

Vom Fernseher zum Monitor

Sie beabsichtigen, einen Monitor zu kaufen? Warum basteln Sie sich nicht einen Monitor aus einem alten Schwarzweiß-Fernsehgerät — und das für weniger als 10 Mark?



In vielen Haushalten existiert neben dem normalen Fernsehgerät häufig auch noch ein kleines tragbares oder ein älteres Schwarzweiß-Fernsehgerät. In den meisten Fällen reicht zum vernünftigen Arbeiten mit dem Computer ein solches Schwarzweiß-Fernsehgerät vollständig aus. Das Problem ist nur der Modulator im C 64, der das niederfrequente Signal des Videoprozessors mit einer hochfrequenten Trägerschwingung mischt beziehungsweise moduliert. Dieses modulierte, hochfrequente Signal liegt am Antenneneingang des Fernsehgerätes an. Im Tuner und im nachfolgenden ZF-Verstärker (Zwischenfrequenz-Verstärker) wird aus dem hochfrequenten Eingangssignal wieder ein Signal erzeugt, das mit dem vom Prozessor erzeugten Signal identisch ist. Durch diese »Signalumsetzerei« werden aber die Flanken des Videosignals abgeflacht. Das Ergebnis ist ein unscharfes Bild. Außerdem wird es durch ein perlschnurartiges Zeilenraster (Moirébildung) beeinträchtigt. Der Modulator des C 64 hat noch eine weitere unangenehme Eigenschaft. Die Trägerfrequenz ist abhängig von der Betriebstemperatur des Computers. Das heißt, daß sich mit zunehmender Temperatur die Frequenz der Trägerschwingung ändert. Das wiederum macht ein permanentes Nachregulieren des Fernsehgerätes erforderlich.

Mit Hilfe eines Kondensators und eines Trimmpotentiometers läßt sich an fast jedes Schwarzweiß-Fernsehgerät eine zusätzliche Monitorbuchse anbringen, die alle Nachteile des Fernsehgerätes beseitigt. Der so entstandene Monitor hat aber nicht nur ein besseres Bild (Bild 1), vielmehr lassen sich auch alle gängigen 80-Zeichenkarten anschließen (Bild 2). Die landläufige Meinung, daß die Auflösung eines Fernsehgerätes mit einer Bandbreite von etwa 5,5 MHz nicht ausreicht, um 80 Zeichen darzustellen, ist falsch. Das möchte ich an einem kleinen Beispiel demonstrieren:

Für 80 Zeichen mit je 8 x 8 Pixel werden 8 x 80 = 640 Pixel benötigt. Nimmt man nun an, daß sich aufgrund des Bildschirmrahmens nur etwa zwei Drittel einer Zeile nutzen lassen, folgt, daß sich die Anzahl der benötigten Pixel pro Zeile um den

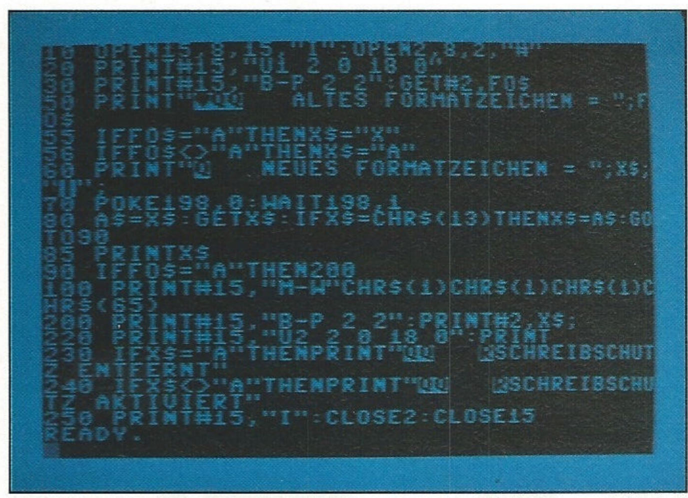


Bild 1. Ein erheblich verbessertes Bild erreicht man mit dem umgebauten Fernseher



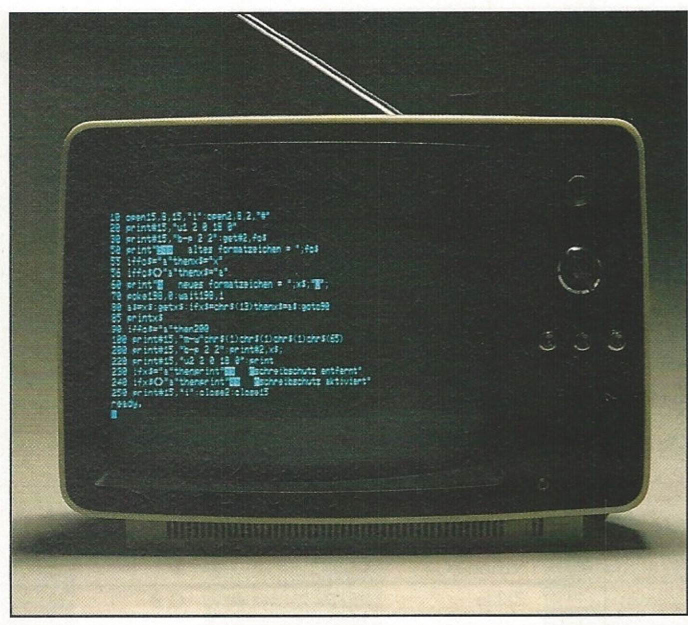


Bild 2. Auch 80 Zeichen pro Zeile sind kein Problem

Faktor 3/2 erhöht. Man braucht also letztendlich 640 x 3/2 = 940 Pixel. Wenn diese Pixel abwechselnd schwarz und weiß sein sollen, muß die Bildröhre mit einer rechteckförmigen Wechselspannung angesteuert werden (Bild 3). Die Frequenz dieser Rechteckspannung läßt sich aus der Anzahl der Schwarzweiß-Sprünge je Zeile und der Zeilenzahl je Sekunde berechnen.
f=940/2*swSprünge/Zeile*625 Zeilen*25 Bilder/Sekunde=7,34 MHz

Die Auflösung

Da in vertikaler Richtung die Schärfe wegen der Unterteilung in Zeilen ohnehin etwas geringer ist, kann auch die Auflösung in horizontaler Richtung etwas geringer sein. Für eine gute Bildqualität reicht daher eine Bandbreite von 5,5 bis 6 MHz voll und ganz aus, auch für die Darstellung von 80 Zeichen.

Doch nun zur Umbauanleitung. Bevor Sie jetzt anfangen, Ihren alten Fernseher zu zerlegen, möchte ich Sie noch warnen. Wenn Ihnen das Wort Elektronik fremd ist, sollten Sie den Umbau anderen überlassen, die mehr davon verstehen. Bei Fehlern können Sie nicht nur Ihren Computer und das Fernsehgerät zerstören, sondern auch Ihrer Gesundheit schaden. Denn im Fernsehgerät sind statische Ladungen auch im ausgeschalteten Zustand vorhanden. Die Spannung, die durch diese statische Ladung zustandekommt, kann durchaus bis zu 15000 Volt betragen.

Was benötigen Sie zu dem Umbau?

Ganz wichtig ist das Schaltbild und nach Möglichkeit ein Lageplan der Bauelemente. Auch Vielfach-Meßinstrumente, Lötkolben und Schraubendreher sollten nicht fehlen. An zusätzlichen Bauteilen werden benötigt:
1 Trimmpotentiometer 10 kΩ
1 bipolarer Kondensator 1 bis 10 µF
1 Anschlußbuchse mindestens dreipolig
1 m abgeschirmtes Kabel

Diese Bauteile sind in jedem Elektronik-Bastelladen erhältlich.

Da die Spannung des Videoausgangssignals am Pin 1 der Video-/Audio-Buchse (Bild 4) 1 V_{ss} beträgt, muß ein entsprechender Anschlußpunkt im Fernsehgerät gefunden werden. Wir haben aber Glück und brauchen nicht lange suchen; denn das Eingangssignal des Videoverstärkers ist auf 1 V_{ss} genormt. Für den nicht so versierten Elektronikbastler ist es natürlich nicht leicht, diesen Punkt zu finden. Darum werden wir uns gemeinsam auf die Suche machen. Bild 5 zeigt das Schaltbild eines kompletten Videoverstärkers inklusive allen erforderlichen Änderungen. Das Schaltbild ist natürlich nur ein Beispiel und wird mit Ihrer Schaltung kaum übereinstimmen. Die charakteristischen Punkte, an denen die Änderungen vorgenommen werden, sind jedoch bei allen Schwarzweiß-Fernsehgeräten identisch, vorausgesetzt, der ZF-Verstärker, Videomodulator und ein Teil des Videoverstärkers sind nicht in einem IC integriert. Sollte das der Fall sein, ist ein Umbau nur mit erheblich höherem Aufwand möglich. Schauen Sie sich nun in aller Ruhe das Schaltbild zu Ihrem Fernsehgerät an und suchen die Bildröhre (das

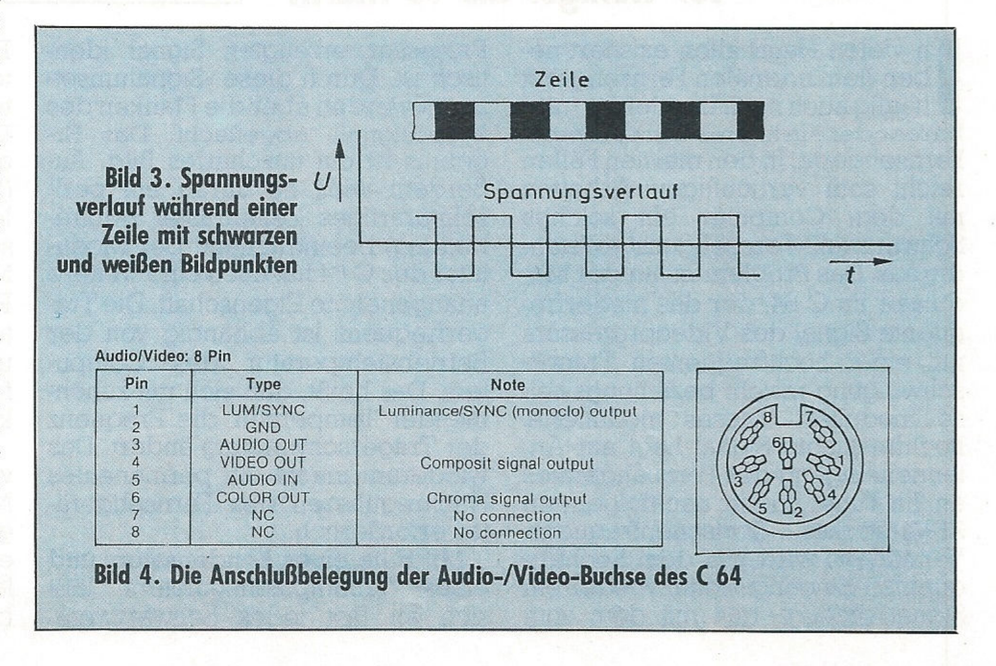


Bild 3. Spannungsverlauf während einer Zeile mit schwarzen und weißen Bildpunkten

Audio/Video: 8 Pin

Pin	Type	Note
1	LUM/SYNC	Luminance/SYNC (monoclo) output
2	GND	
3	AUDIO OUT	
4	VIDEO OUT	Composit signal output
5	AUDIO IN	
6	COLOR OUT	Chroma signal output
7	NC	No connection
8	NC	No connection

Bild 4. Die Anschlußbelegung der Audio-/Video-Buchse des C 64

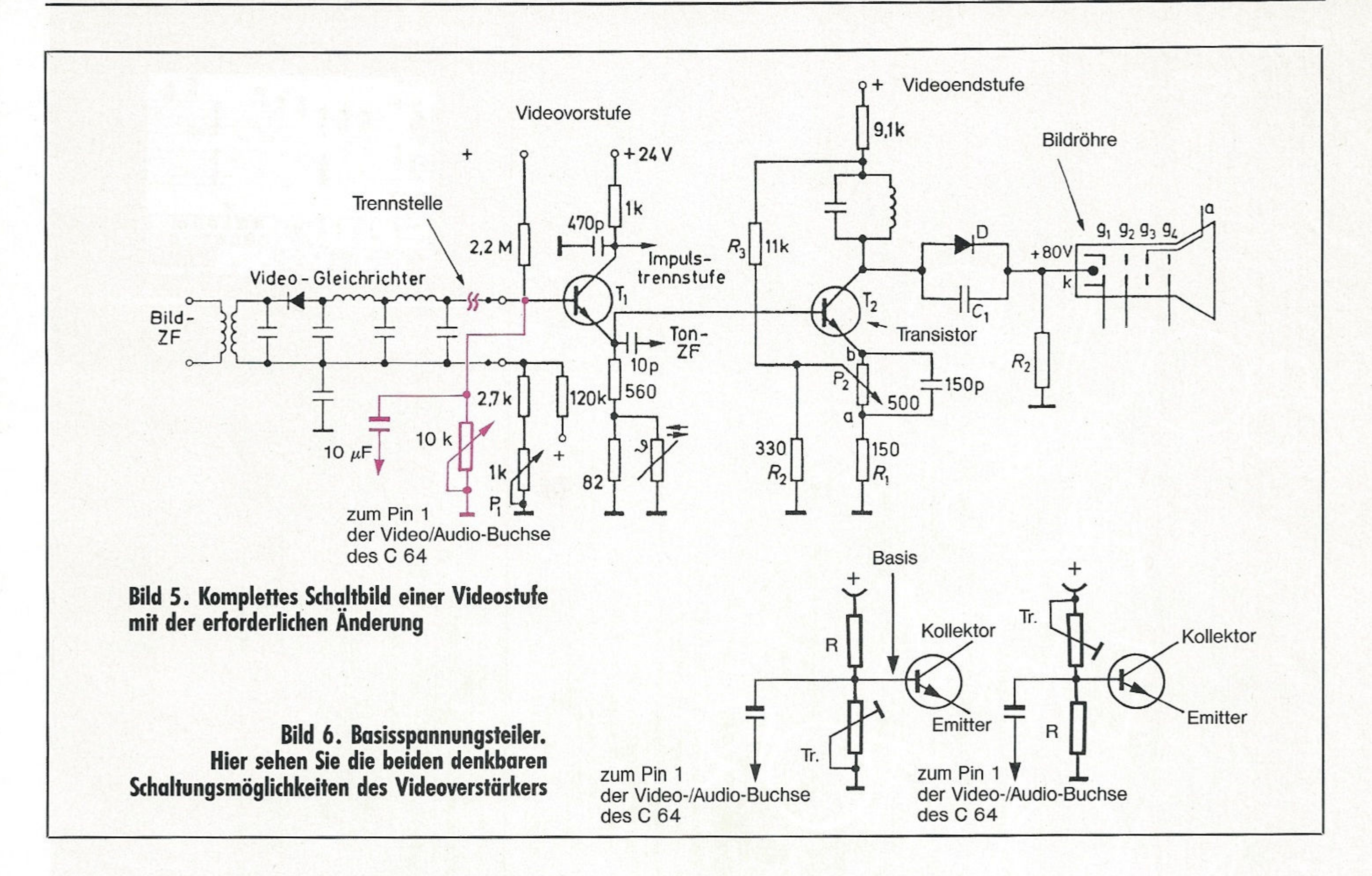


Bild 5. Komplettes Schaltbild einer Videostufe mit der erforderlichen Änderung

Bild 6. Basisspannungsteiler. Hier sehen Sie die beiden denkbaren Schaltungsmöglichkeiten des Videoverstärkers

Schaltsymbol zeigt Bild 5). Haben Sie sie gefunden, geht es weiter. Anschlußpin »k« der Bildröhre (Bild 5) ist nun über einige Bauelemente mit einem Transistor (das Schaltsymbol zeigt Bild 5) verbunden, der Videoendstufe. Die Basis dieses Videotransistors ist wiederum entweder direkt, oder wie gehabt, über einige Bauelemente mit einem weiteren Transistor verbunden, der Videovorstufe. Bei der Basis dieses Transistors handelt es sich um den gesuchten Eingang der Videostufe. An der Basis des Videovorverstärkers befindet sich ein Widerstand, der je nach Fernsehertyp entweder mit »plus« oder »minus« verbunden ist. Er dient zur Einstellung des Arbeitspunktes. Alle Bauteile, die bis auf den Widerstand mit der Basis der Videovorstufe verbunden sind, müssen entfernt werden. Nehmen Sie die Rückwand des Fernsehgerätes ab, vergessen aber nicht vorher den Netzstecker zu ziehen. Suchen Sie die Leiterbahn, die den Widerstand, die Basis des Vorstufentransistors und die restlichen Bauteile miteinander verbindet. Nun ist die Leiterbahn an einer geeigneten Stelle aufzutrennen, so daß nur noch der Widerstand mit der Basis des Vorstufentransistors verbunden ist. Im nächsten Schritt ist das Trimmpotentiometer einzulöten, und zwar so, daß der Widerstand und das Trimmpotentiometer einen Basisspannungsteiler bilden (Bild 6).

Der Umbau

Da das Ausgangssignal des C 64 leicht positiv vorgespannt ist, muß zwischen dem Videoausgang des C 64 und der Basis des Videovorverstärkers ein Kondensator eingebaut werden. Hat man keinen bipolaren Kondensator zur Verfügung, kann ersatzweise ein Elektrolytkondensator genommen werden. Dabei ist jedoch auf die Polarität zu achten, die Sie am besten mit einem Vielfachmeßinstrument ausmessen. Dazu ist der Pin 2 der Audio-/Video-Buchse des C 64 mit der Masse des Fernsehgerätes zu verbinden. Schalten Sie nun beide Geräte ein und messen im Gleichspannungsmodus die Spannung zwischen der Basis des Videovorstufentransistors und dem Videoausgang des C 64 am Pin 1 der Audio-/Video-Buchse. Zeigt das Meßinstrument einen Ausschlag in positiver Richtung, ist der positive Pol derjenige, der mit der roten Klemme des Meßinstrumentes verbunden ist. Dieser positive Pol ist mit dem »+«-Pol des Elektrolytkondensators zu verbinden, nach Möglichkeit aber nicht direkt, sondern über eine geeignete Buchse. Der »—«-Pol des Elektrolytkondensators kommt an die Basis des Videovorstufentransistors. Ist das geschehen, muß mit Hilfe des Trimmpotentiometers noch der Arbeitspunkt des Videovorstufentransistors eingestellt werden. Dazu drehen Sie so lange an dem Potentiometer, bis das Bild einwandfrei steht und nicht mehr verzerrt ist. Möchten Sie auch die Soundmöglichkeiten des C 64 nutzen, läßt sich das ebenfalls machen. Dazu können Sie den Audioausgang des C 64 direkt an den Schleifer des Lautstärkenreglers anlöten.

Das wäre es eigentlich auch schon. Der Umbau hat sich seit einigen Jahren bei mir bewährt. Ich habe außerdem noch einen kleinen Umschalter eingebaut, mit dem ich zwischen Fernseher und Monitor hin- und herschalten kann. (ah)

Es sollten sich nur diejenigen an den Umbau wagen, die mit elektronischen Basteleien vertraut sind.

Durch einen Fehler können Sie nicht nur das Fernsehgerät und den C 64 zerstören, sondern auch Ihrer Gesundheit schaden.



Sie haben einen Monitor ohne Lautsprecher? Dann können Sie sich mit dieser Bastelanleitung einen kleinen Verstärker bauen, mit dem auch Sie die Sound-Qualitäten des C 64 nutzen können.

Viele C 64-Besitzer schließen zur Verbesserung der Bildqualität einen Monitor an ihren Heimcomputer an. Leider fehlt bei den meisten Monitoren ein Tonteil. Der C 64 ist dann seiner sonst so hervorragenden Stimme beraubt. Dieser Zustand ist natürlich nicht haltbar und man wird bald auf Abhilfe sinnen. Aber was ist zu tun? Beim C 64 wird der Ton über die Audio-Video-Buchse nach außen geführt. An diese Buchse müßte also ein externer NF-Verstärker anschließbar sein. Man kann zu diesem Zweck die heimische Stereoanlage hernehmen. Wie dies gemacht wird, wurde bereits im 64'er, Ausgabe 5/85 beschrieben. Auf Dauer ist die Lösung aber nicht befriedigend, denn wer hat schon seinen C 64 immer in Reichweite der Stereoanlage stehen. Eine bessere Möglichkeit ist der Aufbau eines kleinen NF-Verstärkers, der am besten gleich in das Monitorgehäuse eingebaut

wird. Die Anforderungen an diesen Verstärker brauchen nicht besonders hoch zu sein. So dürfte zum Beispiel eine Ausgangsleistung von etwa 1 Watt vollkommen ausreichen. Durch ein großes Angebot spezieller integrierter Schaltungen ist die Realisierung eines solchen Verstärkers denkbar einfach.

Für den in diesem Artikel beschriebenen Verstärker wurde das IC TDA 2002 von SGS-ATES (Bild 1) ausgesucht. Dieses IC wurde für den Einsatz in Autoradios entwickelt.

Deshalb ist es auch mit sehr guten internen Schutzmaßnahmen versehen. Achten Sie aber bitte beim Kauf darauf, daß Sie ein IC mit der Typenbezeichnung TDA 2002 V beziehungsweise H erhalten, denn nur diese Typen haben die internen Schutzmaßnahmen. So können kurzzeitige Überspannungen bis 28 Volt, beziehungsweise Spannungsspitzen bis 40 Volt Ihrem IC nicht schaden. Auch der Schutz gegen thermische und leistungsmäßige Überlastung ist bei dieser Version sehr gut.

Ein weiterer Vorteil des ausgesuchten ICs ist die minimale äußere Beschaltung. Sehen wir uns hierzu die Schaltung (Bild 2) einmal an. Erfahrenen Elektronikbastlern wird auffallen, daß diese Schaltung der Grundschaltung eines nichtinvertierenden Operationsverstärkers entspricht. Bei unserem Verstärker wird mit dieser Beschaltung auch genau der gleiche Effekt erzielt.

Mit den Widerständen R1 und R2 wird der Verstärkungsfaktor eingestellt. Er beträgt v=R1/R2 + 1. Da wir uns auf eine Leistung des Verstärkers von rund 1 Watt beschränken, benötigen wir an einem 8-Ohm-Lautsprecher eine Spannung von 2,8 Volt (effektiv). Der C 64 liefert an seinem Audio-Ausgang eine Spannung von 2,2 Volt. Dies ist ein Spitzenwert, effektiv bleibt davon eine Spannung von 1,1 Volt. Unser Verstärkungsfaktor muß deshalb v=2,8V/1.1V = 2,5 betragen. Von den Widerständen ist R2 mit einem Wert von 22 Ohm vorgegeben. R2 erfüllt zusammen mit dem Kondensator C2 noch eine Funktion zur Störunterdrückung der Versorgungsspannung. R1 errechnet sich somit zu R1=R2*(2.5—1)= 33 Ohm.

Im Prinzip würde die Beschaltung mit den beiden Widerständen ausreichen. Da wir aber einen NF-Verstärker aufbauen wollen, also nur Wechselspannungen verstärken wollen, müssen wir dafür sorgen, daß keine Gleichspannung mit verstärkt wird beziehungsweise unseren Lautsprecher erreicht. Die Folge davon wären nämlich Verzerrungen oder eine Zerstörung des Lautsprechers.

Zur Vermeidung dieser Folgen werden die Kondensatoren C1 bis C3 eingebaut. Der Elko C1 sorgt dafür, daß nur Wechselspannungen den Eingang des ICs erreichen. Da der IC-Eingang sehr hochohmig ist, reicht hier ein kleiner Wert für den Kondensator. Weil intern im TDA 2002 Gleichspannungen vorhanden sind und diese der zu verstärkenden Wechselspannung überlagert werden, wird noch der Kondensator C2 in die Schaltung einbezogen. Er sorgt dafür, daß sich die Verstärkung nur auf die Wechselspannung auswirkt. Zu guter Letzt der Kondensator C3. Er läßt nur Wechselspannung zum Lautsprecher durch. Würde der Lautsprecher nämlich mit einer Gleichspannung beaufschlagt, so würde der unweigerlich durchbrennen.

Wie funktioniert's?

Der Wert des Kondensators C3 sollte mindestens 1000 µF betragen, da sonst die tiefen Frequenzen (sprich die tiefen Töne) nicht bis zum Lautsprecher gelangen.

Das RC-Glied am Lautsprecherausgang dient zur Unterdrückung eventueller Schwingneigungen des Verstärkers.

Ein Glied in der Kette ist bisher unerwähnt geblieben, der Potentiometer P1. Er sorgt dafür, daß wir uns nicht immer alles mit der gleichen Lautstärke anhören müssen. Mit dem Poti wird durch Abschwächung des Eingangssignal eine einfache Lautstärkeeinstellung vorgenommen. Damit wäre die Beschaltung des ICs komplett.

Wenden wir uns deshalb der Schaltung des Netzteiles zu. Die Spannungsversorgung des TDA 2002 darf im Bereich von 8 bis 18 Volt liegen. Da wir nur 1 Watt Ausgangsleistung haben, dürfen wir uns an der unteren Grenze des Versorgungsspannungsbereiches bewegen. Ein Transformator mit 8 bis 9 Volt Sekundärspannung reicht für unsere Zwecke voll aus. Dieser Wert ist nicht kritisch. Achten Sie jedoch darauf, nicht mehr als 20 Volt Sekundärspannung zu verwenden. Denn nach der Gleichrichtung nimmt die Spannung einen etwas höheren Wert an. Strommäßig sollte der Trafo etwa 500 mA liefern können. Dies hängt jedoch auch vom verwendeten Lautsprecher ab. Sollten Sie einen Lautsprecher mit 4 Ohm Impedanz anschließen, so ist entsprechend die Verstärkung abzusenken oder die Leistung des Trafos zu erhöhen. Die Sekundärspannung wird mit Hilfe von vier Dioden des Typs 1N4001 gleichgerichtet. Der Kondensator C4 sorgt anschließend für eine Glättung der Spannung. Ein Wert von 1000 µF ist auch hier die untere Grenze. Der Kondensator C5 sorgt für die Unterdrückung von Störspannungsspitzen der Versorgungsspannung.

Für das IC ist zur Ableitung der entstehenden Wärme ein Kühlkörper erforderlich. Es reicht ein kleiner U-förmiger Kühlkörper.

Der Aufbau des Verstärkers kann auf einer Lochrasterplatine oder besser auf einer nach dem vorgegebenen Layout (Bild 3) erstellten Platine erfolgen. Bild 4 zeigt den Bestückungsplan.

Am günstigsten ist es, diese Platine in den Monitor einzubauen und den Verstärker über den Netzschalter des Monitors dann gleich miteinzuschalten. Da die Monitore in dieser Hinsicht verschieden aufgebaut sind, kann hierzu keine allgemeingültige Einbauanweisung erfolgen. Sollten Sie dies nicht wollen, dann ist dafür Sorge zu tragen, daß die Schaltung oder mindestens das Netzteil in ein gut isolierendes Gehäuse eingebaut wird. Am Transformator liegt nämlich an der Primärseite die volle Netzspannung von 220 Volt an, und diese wirkt bei Berührung absolut tödlich. Aus diesem Grund ist auch beim Aufbau und beim Test des Verstärkers Vorsicht geboten.

Gehen Sie beim Einlöten der Bauteile in die Platine sorgfältig vor. Das IC ist zum Beispiel nicht vor zu starker Hitze von außen geschützt. Am besten ist es, Sie löten die einzelnen IC-Beinchen mit kurzen Abkühlpausen fest. Die richtige Polung der Kondensatoren ist natürlich auch zu beachten. Seien Sie auch vorsichtig beim Testanschluß der Schaltung an den C 64. Ein Kurzschluß der einzelnen Pole am Audio-/Video-Anschluß kann zur Zerstörung des Verstärkers und des Computers führen. Da zum Anschluß des Monitors die Audio-/Video-Buchse belegt ist, muß man einen Adapter oder ein spezielles Kabel für diese Buchse bauen beziehungsweise kaufen. Das übliche Kabel mit einem DIN-Stecker an einer und mit vier Cinch-Steckern an der anderen Seite ist hier zu empfehlen. Bleibt zum Schluß noch zu wünschen, daß Sie allzeit guten Klang haben.

(H. Zwartscholten/ah)

Stückliste:

R1	33 Ohm
R2	22 Ohm
R3	100 Ohm
C1	10 µF/10V
C2	100 µF/10V
C3	1000 µF/25V
C4	100 nF
C5	100 nF
C6	1000 µF/25V
P1	10 kOhm
D1—D4	1N1001
IC1	TDA 2002 V oder H

1 Lautsprecher 1 Watt/8 Ohm
1 kleiner u-förmiger Kühlkörper
1 Trafo 8—9 Volt sek./500 mA

Bauteileliste des NF-Verstärkers

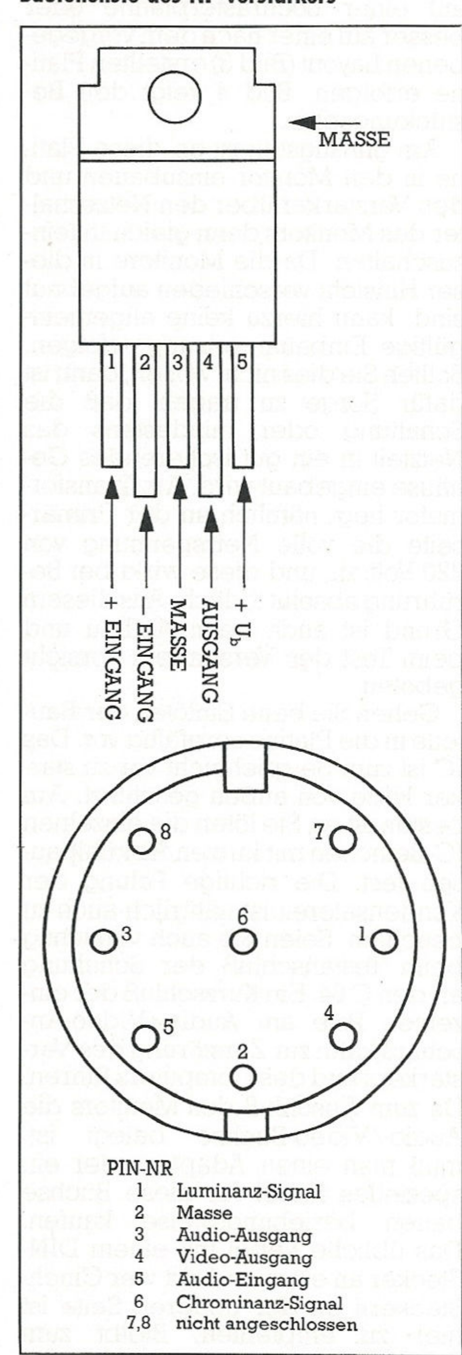


Bild 1. Anschlußbelegung des TDA 2002-Verstärkers und der Audio-/Video-Buchse

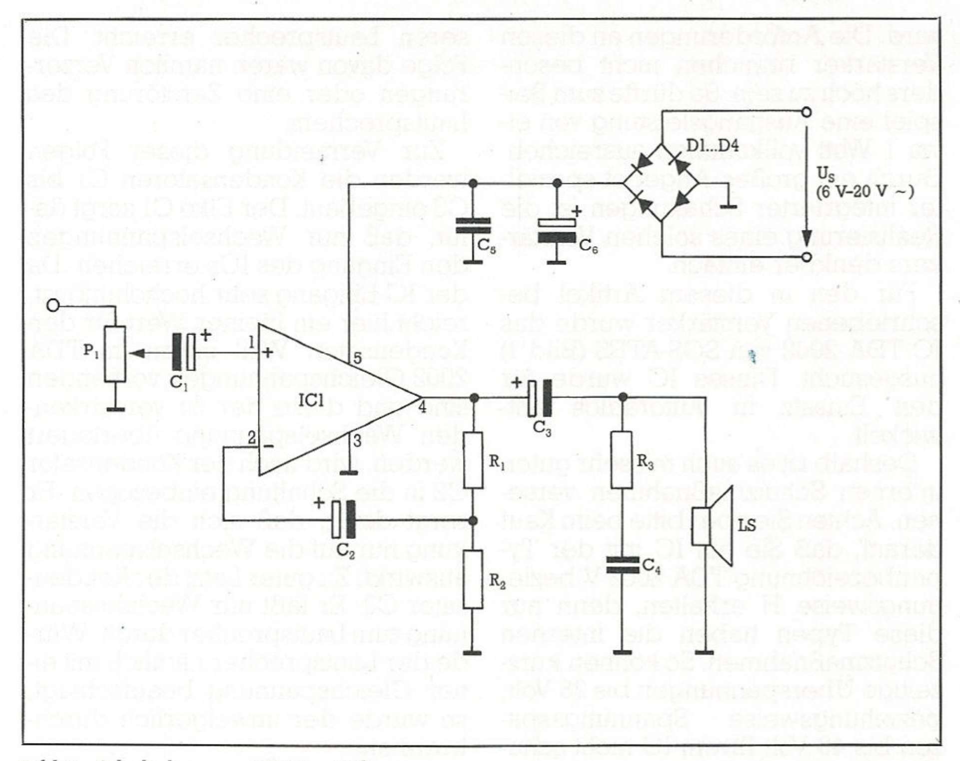


Bild 2. Schaltplan zum NF-Verstärker

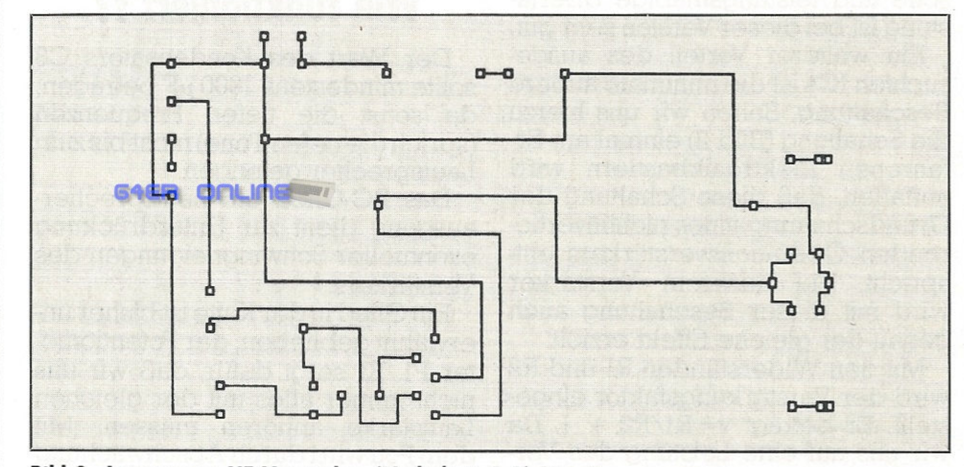


Bild 3. Layout zum NF-Verstärker (Verhältnis 1:1) Lötseite

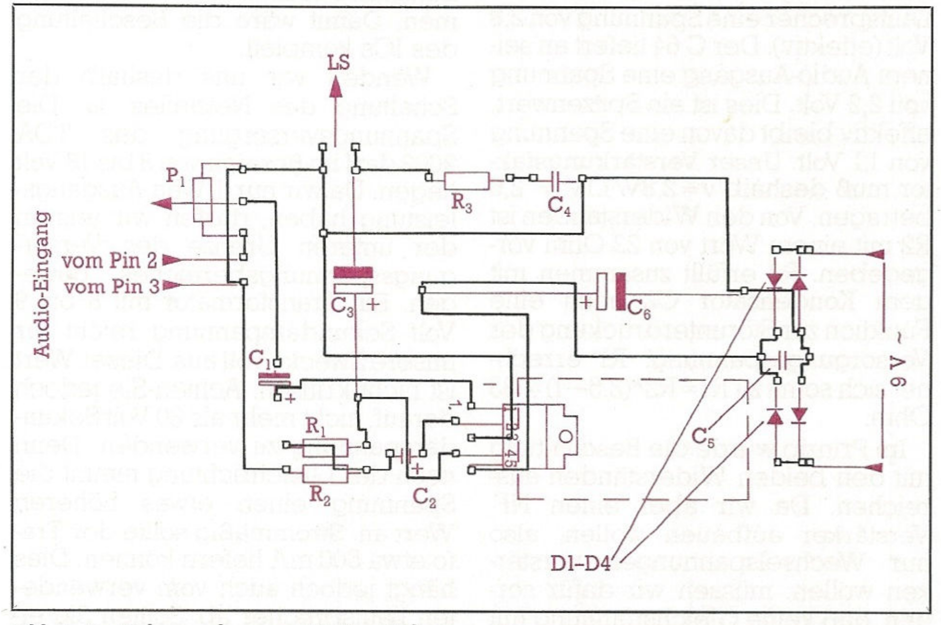


Bild 4. Bestückungsplan zum NF-Verstärker

Simulationen — Spiel und Wirklichkeit

Simulationen sparen Zeit und Geld, sagen die ernsthaften Anwender. Simulationen sind die zeitaufwendigsten und auch teuersten Programme, behaupten die Spieler. Recht haben beide, doch lesen Sie selbst.



Bild 1. Von der Realität kaum zu unterscheiden: Einer der zehn Lufthansa-Simulatoren Quelle: Lufthansa

Was ist eine Simulation eigentlich? Ist es eine computergesteuerte Sub-Realität, die innerhalb unserer realen Welt existiert? Oder haben wir es mit Träumen zu tun, die der Mensch sich auf dem Computer produziert, weil er sich die Realität nicht leisten oder technisch nicht verwirklichen kann? Nun, das ist Ansichtssache. Aber in jedem Fall sind Simulationen Nachahmungen von praktisch oder auch theoretisch existierenden Realitäten. Theoretische Realitäten deshalb, weil man auch Dinge simulieren kann, die es gar nicht gibt, oder von denen man wenig weiß. So wird beispielsweise am Max-Planck-Institut für Astrophysik in Garching bei München an der Simulation einer Supernova-Explosion gearbeitet. Die gesamte Simulation beruht allerdings nur auf Modellen, da die Wissenschaftler noch nicht sehr viel über diese Erscheinung wissen. Simulieren kann man den Vorgang auch nur zu einem minimalen Teil, da der Cray 1B (einer der größten und schnellsten Computer der Welt) nicht genügend Hauptspeicher besitzt. Auf dem Heimcomputer kann man das gleiche feststellen: Computer-Rollenspiele sind nichts anderes als Simulationen von fiktiven Charakteren, die in fiktiven Welten ebenso fiktive Abenteuer erleben.

So weit also zu den theoretischen Simulationen. Man kann natürlich auch real existierende Ereignisse simulieren; die Simulation von in der Praxis bestehenden Dingen ist schließlich auch die häufigere Variante der Simulation. Beispiele sind Flugsimulatoren, Fahrsimulatoren, Wirtschaftssimulatoren, Kriegssimulatoren und so weiter. Eigentlich kann man ja alles simulieren; angefangen vom Vollwaschgang der Waschmaschine bis hin zu Opas Beinbruch ist alles drin. Die Science-Fiction-Literatur läßt sich in dieser Hinsicht auch so einiges einfallen. In Douglas Adams' »Per Anhalter durch die Galaxis« ist die Menschheit auch nur Teil eines Simulations-Experiments, das von viel intelligenteren Wesen (den Mäusen!) überwacht wird. Die Ideen zum Thema Simulation sind so vielfältig wie in keinem anderen Gebiet. Wohl deshalb, weil der Begriff »Simulation« ein sehr breites Spektrum umfaßt.

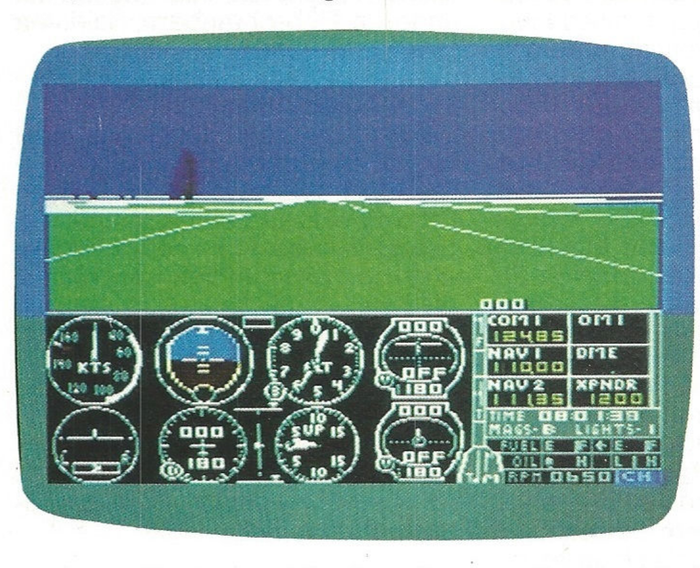


Bild 2. Unrealistischer geht es beim »Flight Simulator II« für den C 64 zu

Den Begriff kann man allerdings in einem engeren Sinne auch sehen, nämlich den der klassischen Arten der Simulation. Also Autofahren, Fliegen, Motorboot fahren — kurz gesagt alles, was sich mit und in Fahrzeugen abspielt. Für unseren Überblick über bestehende Software wollen wir zwischen beiden Begriffsbestimmungen ansetzen, da wir über bestimmte Arten der Simulation bereits ausführlich berichteten (Kriegssimulationen in 64'er, Ausgabe 3/85, Rollenspiele in Ausgabe 9/84) oder es nicht in den Rahmen dieses Magazins paßt (zum Beispiel Computertomographie, Simulationen physischer und chemischer Vorgänge etc. Wer darüber mehr wissen will, sollte Computer persönlich, Ausgabe 2/85 lesen).

Nehmen wir uns mal ein paar Simulationen vor und vergleichen sie mit den eingangs gemachten Behauptungen. Simulationen sparen Geld und Zeit, sagten wir. Auf den »Flight Simulator II« für unseren Heimcomputer trifft das ganz be-



Bild 3. Fantastische Grafik bietet sich den Testfahrern des Mercedes-Benz-Simulators
Quelle: Mercedes-Benz

stimmt nicht zu, denn unser Heimanwender setzt sich — im Gegensatz zu Ballerspielfans — schon mal ein paar Stunden mehr an den Computer. Und mehr kosten als ein Ballerspiel tut der Flugsimulator allemal. Wenden wir diese Behauptung allerdings auf eine ernsthafte Simulation an, dann sieht die Sache schon ganz anders aus. In der Forschung beispielsweise kann man nicht jedes Gedankenmodell in ein reales Modell umsetzen. Die Computertechnik spart hier viel Zeit und Geld, wenn man mit Hilfe einer Simulation die »Was passiert, wenn ...«-Frage untersucht. Beispiel: Zugunglücke im realen Modell durchzuführen, würde nicht nur viel kosten, sondern wäre für eine ganze Menge Leute ziemlich ungesund. Um ihre Lokführer für solche Notfälle zu trainieren, hat die Western Railroad Company in den USA einen »Railroad Simulator« gebaut. Dabei ist im Unglück auch das Gewicht der Passagiere und der Ladung einberechnet. Ähnliches macht auch die Lufthansa mit ihren Flugsimulatoren. In der Wirtschaft kann durch Simulationen und daraus folgenden Trendrechnungen die Realisierung eines geplanten Umsatzes im Voraus als unmöglich erkannt werden. In der Fahrzeugindustrie braucht man nicht Unmengen von verschiedenen ABS (Anti-Blockier-Systemen) zu bauen, um die beste Konstruktion durch Tests zu finden. Hier wird einfach simuliert, und die in der Simulation am besten abschneidende Lösung wird realisiert. Man könnte noch viele Beispiele in dieser Richtung finden.

Welche Simulationen gibt es nun für unsere Heimcomputer, und was unterscheidet sie von den ernsthaften, in der Praxis angewendeten Simulatoren? Eigentlich eine ganze Menge: Die Realitätsnähe, das benutzte Computersystem, der Preis und der Nutzen. Auf dem Heimcomputer haben wir es zu 95 Prozent mit Spielen zu tun. In der Praxis gibt es keine Spiele. Wer mit einer Cray-Anlage oder einem großen Luftkampfsimulator (zum Beispiel Thomson oder Sogitec) zu spielen beginnt, bekommt entweder Hausverbot, eine Kündigung oder eine saftige Rechnung in Millionenhöhe. Aber sehen Sie selbst in unserem kleinen Überblick, was es da noch alles gibt.

Flugsimulationen

Die ersten Flugsimulatoren entstanden in den 40er Jahren. Sie dienten nur als Navigationstrainer für den Instrumentenflug. Ende der 50er Jahre wurden daraus die sogenannten CSS (Cockpit System Simulatoren), mit denen man schon rechnerisch alle Funktionen eines Flugzeugs simulieren konnte — damals noch mit Analogrechnern, die im Platzverbrauch »unschlagbar« waren. Mit der zunehmenden Miniaturisierung und Fortschritten in der Digitaltechnik kamen dann immer bessere Simulatoren. Dabei war die Vorbereitung der ersten Mondlandung ein großer Schrittmacher für die technischen Verbesserungen. Mitte der 70er Jahre kamen dann die ersten Flugsimulatoren der heutigen Generation auf, die sich im wesentlichen auf drei Baugruppen stützten: Die Nachbildung des Cockpits, die Projektionseinrichtung, die fortlaufend die Bilder vor den Cockpitfenstern erscheinen läßt, und das (hydraulisch gesteuerte) Bewegungssimulationsgerät. Die Lufthansa betreibt schon zehn Simulatoren (Boeing und Airbus), von denen das neueste Exemplar immerhin 23 Millionen Mark gekostet hat. Eine Flugstunde mit einem echten Flugzeug kostet etwa 39000 Mark, der Simulationsbetrieb nur 1400 Mark pro Stunde. Bei den Heimcomputerspielen sieht das ganz anders aus: Man zahlt einmal und kann unendlich lange fliegen. Es ist halt nicht ganz so realistisch. Die wichtigsten Flugsimulatoren für den C 64:

Flight Simulator II

Der »Flight Simulator II« (Bild 2) von SubLogic ist das Optimum an Flugkomfort, vergleicht man ihn mit anderen Simulationsspielen. In der Verpackung findet man zwei 100 Seiten starke Handbücher, von denen sich das eine mit der Bedienung des Programms befaßt, und das zweite eine Art Flugkurs (auch für echte Flugzeuge) ist. Der »Flight Simulator II« ist ein Echtzeitsimulator, zeichnet sich durch hervorragende Grafik und eine starke Realitätsnähe der Instrumentenanzeigen und des Flugverhaltens aus. Ein Muß für den Flugfan.

Super Huey

Ein Hubschraubersimulator, der grafisch gut aufbereitet ist. Die elektronischen Anzeigen des Originalvorbildes (UH-1X) sind relativ wirklichkeitsnah dargestellt. Das nicht ganz so realistische Flugverhalten des Helikopters wird durch ein besonderes elektronisch gesteuertes Stabibilisationssystem erklärt, das angeblich auch das Vorbild haben soll. »Super Huey« ist ein durchaus noch zu empfehlendes Spiel, das aus einer relativ realitätsnahen Flugsimulation und ziemlich unrealistischen Spielelementen (Kampf mit anderen Hubschraubern, Soldaten retten etc.) besteht.

Andere Flugsimulationen

Von Microprose gibt es den »Solo Flight«, ein mittelprächtiges Simulationsspiel. Daneben gibt es noch etliche unrealistische Programme wie beispielsweise »F15 — Strike Eagle«, »Skyfox«, »Fighter Pilot« oder »ACE«. Mit dem demnächst zu erwartenden »Jet« von SubLogic dürfte allerdings wieder etwas Gutes in dieser Richtung auf den Markt kommen. Wer einen ganzen Flughafen (nicht das Flugzeug) kontrollieren will, dem sei »Kennedy Approach« empfohlen, ein gutes Spiel, das den Spieler ständig in Aktion hält.

Fahrsimulatoren

Auch im Kraftfahrzeugbau hat sich die Simulation als wichtige Komponente erwiesen. Man mußte drei Komponenten in eine Simulation bringen: Fahrer, Fahrzeug und Straße. Dies wurde anfangs noch mit Analogcomputern durchgeführt.

Mit einem Patch-Board wurde eine Simulation zusammengestrickt, mit der immer nur eine spezielle Situation untersucht werden konnte. Für eine neue Situation trat dann ein zeitraubender Umbau ein. Die Straße wurde durch den Blick einer Minikamera simuliert, die über die aufgebaute Miniaturlandschaft entsprechend den Bewegungen des Lenkrades folgte. Inzwischen gibt es wesentlich bessere Simulatoren, die alle digital arbeiten. VW, Mercedes und BMW benutzen beispielsweise sehr professionelle Geräte. Im VW-Simulator wird jedes einzelne Rad mit APUs (Hilfsprozessoren) simuliert. Auch der Simulator von Mercedes liefert erstaunliche reale Fahrerlebnisse. Er ist in Bild 3 zu sehen Nicht so realistisch, doch durchaus überzeugend, sind die Fahrsimulationsspiele. Angefangen hat es mit »Night Driver«, einem Spielautomaten, der nur Gas und Lenkrad hatte, und auf dem Bildschirm links und rechts Pfosten anzeigte, weiter nichts. Inzwischen gibt es Spielhallen-Automaten, die kaum noch in der Grafik zu schlagen sind. TX-1, ein Formel-1-Simulator, hat beispielsweise eine drei Bildschirme große Grafik. Auch vor dem Heimcomputer hat diese Entwicklung nicht Halt gemacht. Die interessantesten Programme in dieser Richtung:



Bild 4. »Pitstop II« ist eine Formel 1-Simulation für zwei »Fahrer«

Revs

Das brandneue »Revs« ist die erste Autorenn-Simulation, die den Begriff »Simulation« voll verdient. Hier kann man mit einem Formel 3-Wagen über die »millimetergenau« nachgebildeten Strecken von Brands Hatch und Silverston rasen. Vom Verhalten des Motors, der Lenkung und der Straßenlage bis hin zur Aerodynamik stimmt einfach alles — deswegen ist »Revs« auch besonders schwer zu spielen. In einer der nächsten Ausgaben werden wir »Revs« genauer vorstellen.

Pit Stop II

»Pit-Stop II« (Bild 4) ist eine Fahrsimulation für zwei Spieler gleichzeitig. Die Grafik ist umwerfend, das Spiel macht Spaß. Nach einiger Zeit kann es jedoch zu Verkrampfungen der Joystickhand kommen. In der Simulation sind auch Reifenabnutzung und Benzinverbrauch berücksichtigt.

Weitere Fahrsimulationen

Von Commodore gibt es zwei als Module angebotene Fahrspiele: »Road Race« und »Le Mans«. Das erste ist ein Nachbau des alten Night Driver, das zweite ist ein Paddle-gesteuertes Spiel mit Vogel-Perspektive. Beide Spiele sind im Ausverkauf in den Kaufhäusern zwischen 5 und 15 Mark erhältlich. Ein Klassiker ist der Spielhallen-Hit »Pole Position« der aber nicht an »Pitstop II« herankommt. Demnächst soll es »Pole Position II« geben, über dessen Qualität noch niemand etwas sagen kann (die Ankündigung war schon vor sechs Monaten).

Wirtschaftssimulationen

Simulationen aus dem kaufmännischen Bereich können in der Finanzplanung, Preiskalkulation oder bei Börsen-Geschäften helfen. Preiskalkulationen werden mit Spreadsheets (= Tabellenkalkulationsprogrammen) durchgeführt. Bei der Änderung eines Wertes im Spreadsheet passiert eigentlich nichts anderes als eine »Was wäre, wenn...«-Simulation. Spielerisch kann man das natürlich auch auskosten. Neben den Tabellenkalkulationen kann man auf dem C 64 folgende Simulationen erleben:

M. U. L. E.

Die beste Wirtschaftssimulation, die es im Augenblick für den C 64 gibt. Sie landen auf dem Planeten Irata, auf dem Sie mit Landwirtschaft, Energie- und Erzabbau Geld machen müssen. Muß man unbedingt haben, wenn man lange und viel Spaß haben will. Mit bis zu vier Spielern gleichzeitig.

Millionaire

Blue Chip Software, die Spezialisten für Wirtschaftsspiele, haben mit Millionaire ein gutes Börsenspiel entwickelt. Die C 64-Version ist im Gegensatz zu anderen Versionen grafisch mißlungen. Dafür aber tut sich spielerisch einiges. Bis zu 14 Leute können mitspielen.

Kaiser

Sie sind im Jahr 1700 Regent eines deutschen Kleinstaates. Eine Simulation der Zusammenhänge wirtschaftlicher, sozialer und militärischer Faktoren. Spielerisch und grafisch nicht übel; die einzige deutschsprachige Simulation, die es derzeit gibt. Bis zu sechs Spieler.

Tycoon

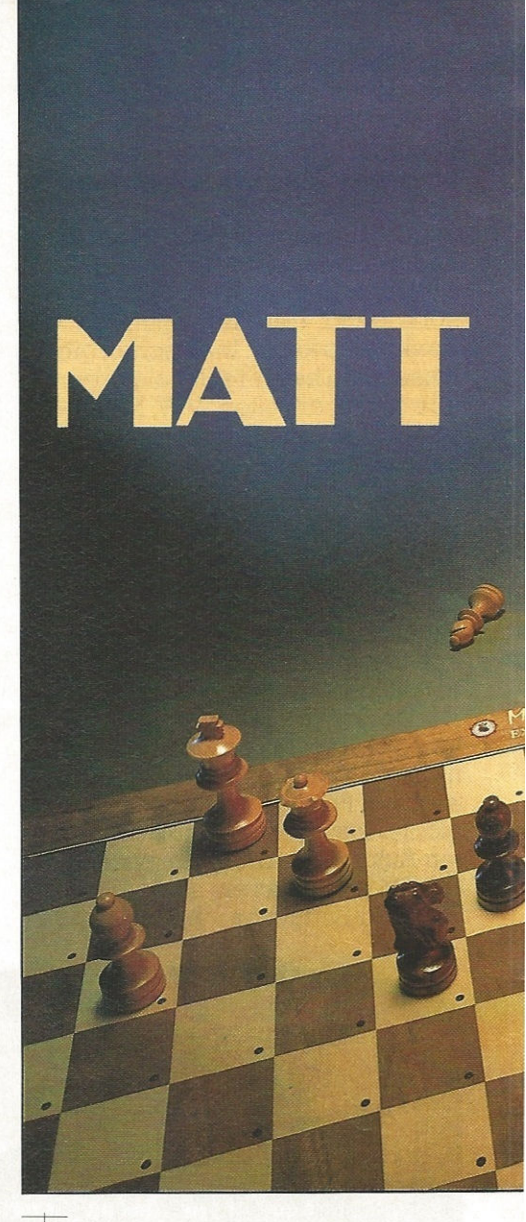
Ein Spiel, in dem es um Handel geht. Der Spieler muß jeweils in die günstigsten Waren investieren und mit bestmöglichem Gewinn verkaufen. Bis zu 14 Spieler.

Was kommt als nächstes?

Man kann natürlich noch wesentlich mehr simulieren als hier angegeben, und auch die Anzahl der erhältlichen Spiele ist riesig. Wir haben nur die interessantesten herausgepickt. In Zukunft wird sich wohl noch so einiges tun. Wer »Radar Raiders« auf dem Amiga gesehen hat, will den »Flight Simulator II« in die Ecke schmeißen. Wer von »Pole Position II« schwärmt, muß leider vorerst noch in die Spielhalle gehen. Und die Zukunft wird sicher noch so einiges bringen, von dem wir heute noch nicht mal zu träumen wagen. Wer mit einem Flugsimulator der nächsten Generation abstürzt, wird vielleicht wegen der Echtheit der Simulation eine dieser Tüten, die in jedem Flugzeug Grundausstattung sind, benötigen. Hoffen wir das Beste für unsere mutigen Piloten und Chauffeure, ob nun am Heimcomputer oder in der Realität.

(Manfred Kohlen/bs)

VIERMAL SCHACH MATT (TEIL 2)



In einem spannenden Turnier haben wir vier Schachprogramme für den C 64 gegeneinander antreten lassen. Außerdem ließen wir sie gegen den Weltmeister der Schachcomputer, das deutsche Seriengerät »Mephisto« spielen.

In der letzten Ausgabe wurden vier Schachprogramme für den C 64 vorgestellt: »Grandmaster«, »Chess 7.0«, »Sargon III« und »Colossus 4.0«. In der Zwischenzeit haben wir einen Wettkampf veranstaltet, in dem wir alle Programme unter Turnierbedingungen gegeneinander spielen ließen. Die normalen Turnierbedingungen sehen so aus, daß in einer Partie 40 Züge in zwei Stunden ausgeführt werden müssen. Da praktisch jedes Schachprogramm eine Turnierstufe anbietet, haben wir unter diesen Voraussetzungen spielen lassen. Außerdem haben wir alle Schachprogramme so »intelligent« wie nur möglich gelassen. Das heißt, daß »Sargon III« und »Colossus 4.0« die Bedenkzeit des Gegenspielers für eigene Berechnungen ausnutzen durften. Zusätzlich wurde die sehr umfangreiche Eröffnungsbibliothek von »Sargon III« nicht abgeschaltet. Das verschaffte manchen Programmen zwar von vorneherein große Vorteile über die Gegenspieler. Allerdings wird ein guter menschlicher Spieler ja in einem Turnier auch nicht ein paar Stufen heruntergeschaltet, nur weil er einen etwas schwächeren Gegner hat.

Das Ergebnis

Kommen wir nun zu den Ergebnissen des Turniers, die wir in einer Tabelle zusammengefaßt haben. Daraus geht klar hervor, daß »Grandmaster« das schwächste der getesteten Programme war. »Chess 7.0« bestreitet einsam das Mittelfeld. Am spielstärksten haben sich »Sargon III« und »Colossus 4.0« erwiesen.

Die Fronten waren dabei sehr klar — bis auf den Gleichstand zwischen »Sargon« und »Colossus« ist das Ergebnis eindeutig.

Dieses Ergebnis kann man nun auch ein wenig analysieren. Daß »Grandmaster« so schlecht abschneidet, ist gar nicht so verwunderlich. Das Programm ist jetzt knapp vier Jahre alt und ist somit nicht auf dem neuesten Stand der Schach-Programmierung. Die anderen drei Programme sind wesentlich jünger und können mit ihren neuen Algorithmen effektiver spielen.

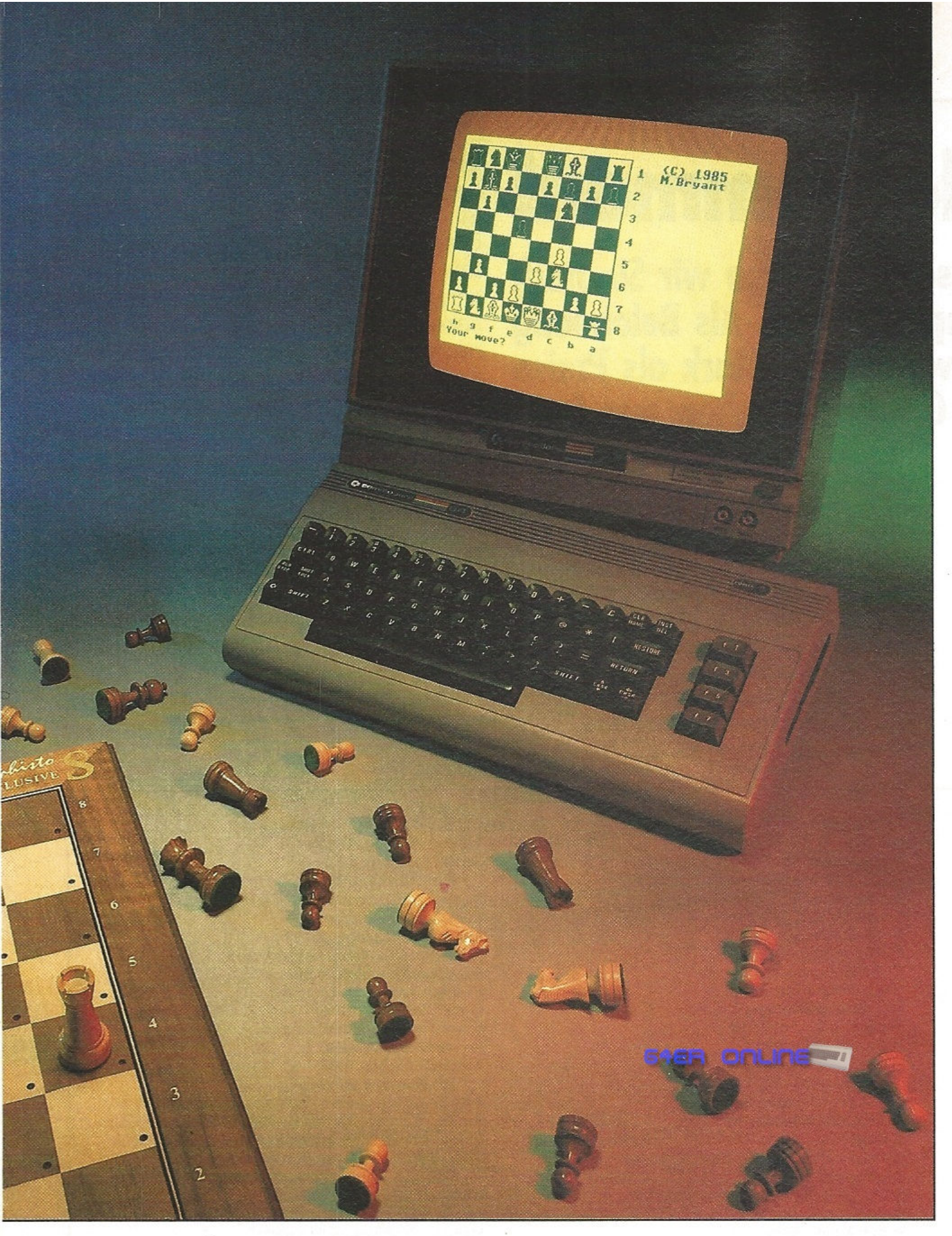
Daß sich »Sargon III« und »Colossus 4.0« gegenüber »Chess 7.0« durchsetzen konnten, liegt sicher an deren Möglichkeit, die Bedenkzeit des Gegeners für eigene Berechnungen zu nutzen. Beide berechnen während der normalen Bedenkzeit den besten Zug des Gegners. Auf dieser Basis führen sie dann schon Überlegungen für den nächsten Zug durch, während der Gegner noch über seinen nachdenkt.

»Sargon III« kann schließlich noch mit seiner riesigen Eröffnungsbibliothek in der ersten Spielphase so viele Vorteile herausholen, daß im weiteren Verlauf des Spiels die anderen Schachprogramme stets im Nachteil sind und nur mit Glück und besonders guten Zügen gegen »Sargon III« widerstehen können. Deswegen verwunderte der Erfolg von »Colossus 4.0« gegen »Sargon III« sehr. Die allgemeine Spielstärke von »Colossus 4.0« muß wohl, sieht man von der sehr viel kleineren Eröffnungsbibliothek ab, größer als die von »Sargon III« sein.

Zu guter Letzt sei gesagt, daß selbst ein Turnier von jeweils vier Partien der Programme gegeneinander noch keine absoluten Aussagen über die Spielstärke machen kann. Die festgestellten Tendenzen sind aber eindeutig.

Die Herausforderung: Mephisto

Um das Turnier abzurunden, wollten wir einen »echten« Schachcomputer mitspielen lassen. Aus Zeitgründen mußten wir uns aber auf einige Partien gegen »Sargon III«, »Colossus 4.0« und »Grandmaster« be-



schränken. Als Schachcomputer haben wir uns dasjenige Gerät herausgesucht, das von Spezialisten als das stärkste Seriengerät bezeichnet wird: »Mephisto« von der deutschen Firma Hegener & Glaser. Bei den Partien gegen »Grandmaster« fegte »Mephisto« seinen Gegenspieler in weniger als dreißig Zügen vom Brett. Etwas länger konnten sich die Testsieger »Sargon III« und »Colossus 4.0« halten. Aber auch sie mußten sich »Mephisto« eindeutig geschlagen geben. Ein auf Schach spezialisierter Computer ist also sehr viel spielstärker als ein »Universalcomputer« wie der C 64 mit einem guten Programm. Denn während der »Mephisto« eben nur Schach spielen kann, ist der C 64 ganz nebenbei auch als Textverarbeitungssystem oder als Lernmittel zur Programmierung einsetzbar.

Verblüffend ist, daß C 64 wie Mephisto vom technischen Standpunkt sehr ähnlich sind. Beide verwenden fast denselben Prozessor und sind somit ähnlich programmiert — allerdings ist der Mephisto von vorne herein etwas schneller und muß sich dann auch nicht mit Bildschirmgrafik und Soundeffekten herumschlagen wie der C 64. Somit hat er intern viel mehr Zeit für seine Berechnungen.

Was für wen?

Die klare Folgerung: Wer sehr viel, sehr gerne und sehr gut Schach spielt, ist mit einem guten Schachcomputer besser bedient, auch wenn dieser ein paar Mark mehr als ein Schachprogramm kostet. Wer nur gelegentlich spielt oder sich nicht für einen Meisterspieler hält und auch noch einen C 64 sein eigen nennt, der wird mit einem guten Schachprogramm zufrieden sein. Schließlich bieten manche Schachprogramme einiges mehr an Komfort wie ein Schachcomputer, so zum Beispiel Abspeichern von Partien oder grafische Darstellung. Absolute Schachfans brauchen dann sowieso beides: Schachprogramm und Schachcomputer. (bs)

Info: Grandmaster: Kingsoft, Fritz Schäfer, Schnakebusch 4, 5106 Roetgen, 39 Mark (Kassette), 49 Mark (Diskette)
Chess 7.0: Softline, Rut Alverdes, Schwarzwaldstr. 8a, 7602 Oberkirch, 198 Mark (Diskette)
Sargon III: Funtastic, Tannhäuserpl. 22, 8000 München 81, 182 Mark
Colossus 4.0: Rushware, An der Gümpgesbrücke 24, 4044 Kaarst 2, 39 Mark (Kassette), 59 Mark (Diskette)
Mephisto: Hegener + Glaser, Arnulfstr. 12, 8000 München 2

weiß \ schwarz	Grandmaster		Chess 7.0		Sargon III		Colossus 4.0		
Grandmaster	—	—	1	0	0	0	0	0	+1
Chess 7.0	1	1	—	—	.5	1	0	1	+4.5
Sargon III	1	1	1	1	—	—	1	.5	+5.5
Colossus 4.0	1	1	1	.5	.5	1	—	—	+5
	-6		-4.5		-3		-2.5		

Gesamtergebnis:

Grandmaster	-5
Chess 7.0	0
Sargon III	+2.5
Colossus 4	+2.5

Bei unserem kleinen Turnier spielte jedes Schachprogramm viermal gegen jeden Konkurrenten, wobei jedes Programm genau zweimal schwarz und zweimal weiß erhielt. Bei Gewinn gab es einen Punkt, bei Verlust keinen und bei Remis jeweils einen halben. Die Gewinnpunkte für Weiß wurden positiv, die Verlustpunkte (das heißt Gewinnpunkte der Gegner) für Schwarz negativ gezählt und addiert. So errechnen sich die Gesamtergebnisse.

Die Spiele kommen!

In unserer September-Ausgabe fragten wir Sie nach Ihrer Meinung über Computerspiele. Als Belohnung war Software im Wert von 10000 Mark als Preise ausgesetzt. Die Gewinner stehen nun fest.

Genau 150 Leute in Deutschland werden sich dieser Tage freuen, denn jeder von ihnen hat ein Programm gewonnen. In diesem Sinn möchten wir uns nochmals recht herzlich bei den Stiftern der Programme bedanken. Die Programme sind schon unterwegs und müßten dieser Tage bei den Gewinnern eintreffen.

Mit diesem Preisausschreiben war auch unser Fragebogen verbunden, in dem wir Sie über Ihre Meinung über Spiele im Allgemeinen und unsere Spiele-Tests im Besonderen baten. Wir bedanken uns für die ausführliche, interessante und aufschlußreiche Beantwortung dieser Fragen. Wir freuen uns ganz besonders darüber, daß Sie fast einstimmig unseren Spiele-Teil »gut« oder sogar »sehr gut« einstufen. Dem Wunsch nach noch mehr Tests werden wir wohl nicht nachkommen, da wir uns jeden Monat nur die besten Spiele herauspicken wollen. Sie wollten auch bessere Vergleichsmöglichkeiten bei den Spielen. Diesem Wunsch haben wir schon vor zwei Ausgaben durch unsere Neukonzeption des Spiele-Teils nachgegeben. Daß wir mit unseren Bewertungskästen ziemlich richtig liegen, zeigten Ihre Antworten auf die Frage, was Ihnen bei einem Spiel am wichtigsten sei. Grafik, Sound und Spielidee sind bei Ihnen am gefragtesten.

Der Spieleteil wird also in der nächsten Zeit von uns so weitergeführt werden, wie Sie ihn auch in dieser Ausgabe vorfinden. Das soll Sie aber nicht davon abhalten, uns weiterhin Ihre Anregungen und Kritik mitzuteilen, auch wenn es dann keine Programme zu gewinnen gibt.

(bs)

Wir möchten uns bei folgenden Firmen für die zur Verfügung gestellten Preise bedanken: Ariolasoft, FunTastic, Imagine, Joysoft, Kingsoft, Markt & Technik, Mastertronic, Micro Händler, RushWare

Folgende 150 Personen haben je ein Programm für den Commodore 64 gewonnen:

Roy Leschinsky, Hagen
Björn Kunze, Darmstadt
Reinhard Gezeck, Nusse
Roger Meyer, Luxemburg
Claus Elsner, Aschaffenburg
Sascha Seifert, Gundelsheim
H. Ditzel, Vlissingen (Holland)
Jan Struck, Kiel
Klaus Sopp, Elmshorn
Manfred Mebus, Essen
Manfred Schattschneider, Eppelborn
Jörg Gatzemeier, Hannover
Jörg Sawallek, Gelsenkirchen
Michael Langer, Wolfsburg
Oliver Haupt, Rodenbach
Karsten Geese, Burgwedel
Steffen Bastek, Neckarbischoffsheim
Guido Mengelberg, Wuppertal
Rainer Korzensdorfer, Hanau
Roland Friedrich, Vöhl
Thorsten Weiß, Deizinau
Jörg Stührwohlot, Heide
Mustafa Allbayrakdar, Karlsruhe
Karl-Horst Schmitt, Mühlheim
Oliver Winkler, Leimen-Gau
Ulrich Wiedemann, Kornwestheim
Michael Maier, Dietfurt
Wolfgang Wolter, Hilden
Walter Schock, Davos (Schweiz)
Peter Börner, Nürnberg
Dirk Kurt, Düsseldorf
H.D. Brandt, Bad-Meinberg
Frank Hermann, Düsseldorf
Klaus Gottke, Düsseldorf
Uwe Tröger, Goslar
Michael Kugler, Schwaig
Georg Maier, Tuntenhausen
Karin Zehetmayer, Köln
Bernhard Nullock, Frankfurt
Bernd Lautenschlager, Nürnberg
Hans Zagorski, Gütersleben
Oliver Müller, Ottobeuren
Thomas Rappelt, Würzburg
Reiner Kopp, Michelstadt
Peter Stendel, Brombachtal
Ralf Müller, Blankenheim/Wald
Rolf Kolodzey, Marl
Achim Jeßberger, Kreuzwth.-Röttbach
Amgar Baumann, Aalen
Kraus Gerhard, Tirschenreuth
Gerhard Klingele, Tunau
Matthias Heller, Pohlheim
Christian Busch, Merzhausen
Holger Ickels, Hamburg
Martin Langenbacher, Tübingen
Michael Stöhr, Frankfurt
Dirk Farin, Tübingen
Michael Lehner, Eppendorf
Harry Hellmann, Wolfsburg
Martin Sige, Deggendorf
Carten Hirsch, Schneverdingen
Rainhard Plohmann, Glücksburg
Burkhard Weichler, Nautorf
Hans-Joachim Knittel, Stuttgart
Marco Schlicht, Berlin
Horst Hahn, Amberg
Robert Kunde, Köln
Harald Schröder, Bad Oldeslohe
Thorsten Kramer, Lauenburg
Heinz Musche, Velbert
Christian Gaal, Oberwart
Olaf Przibilla, Munster
Thomas Dücker, Hamburg
D. Stattenberger, Bremen
Bernhard Müller, Riedholz (Schweiz)
Dirk Ulinksiek, Vlotho
Michael Schreyer, Nürnberg
J. Schröder, Herten
Guido Heinze, Brühl Vochem
Ernst-Dieter Stacke, Uetersen
Peter Jäger, Landau
Alex Dohmen, Wasserberg
Jürgen Burger, Winden
Ralf Achatz, Dortmund
Andreas Henfe,Menden
Michael Schramm, Hamburg
Christian Leopold, Bad Bodendorf
Nis Grella, Rendsburg
Rolf Beier, Ahlen
Matthias Plohmann, Porta-Westfalica
Thomas Veit, Mainz
Axel Dröge, Bestwig
Hartmut Dittmer, Kirchgellensen
Michael Groß, Mainz
Drik Struebig, Duisburg
Andreas Blust, Karlsruhe
Otto Martin, Ochsenthal
Adelberger, Tirschenreuth
Matthias Marciniak
Ulrich Doeker, Köln
Alexander Sarras, Wien (Österreich)
Karsten Schmidt, Berlin
Michael Eberle, Wil (Schweiz)
Klaus Kruase, Oberwallmenach
Ch. Bauermeister, Bad Berleburg
Winfrid Przibilla, Munster
Sörke Nickelsen, Raisdorf
Heribert Volkmer, Weil
Markus Zimmermann, Zürich (Schweiz)
Christian Setzer, Rödental
Andre Wehn, Biedenkopf
Joachim Bieber, Bergkleinfeld
Andreas Ullmann, München
Christof Oswald, Braschhofen
Frank Diesel, Straubing
Oliver Faust, Kapellen (Belgien)
Jürgen Bönig, Moltsee
Alexander Schon, Trier
Patrick Warmbier, Sereetz
Reto Bryan von Gunten, Selfingen (Schweiz)
Hartmut Eumann, Worms
Wener Riecke, Bremgarten
Georg Wilcke, Filderstadt
Daniel Lindenmann, Wittnau (Schweiz)
Michael Larisch, Mainlens
Marco Fischer, Aschaffenburg
Michael Cassemeyer, Troisdorf
Marc Schmitter, Esslingen
G. Schöning, Marburg
Gernot Zelzer, Graz (Österreich)
Günter Jonczyk, Köln
Anton Stupin, Düsseldorf
Thorsten Pelka, Hehlen
Hauke Walter, Gmunden (Österreich)
Christian Märkl, Zeitlarn
Günther Haupt, Markt Schwaben
Carsten Scharf, Solms
Frank Scharfe, Ludwigshafen
Thomas Rank, Ellhoven
Jürgen Anton, Karlsruhe
Oliver Weindl, Rottenburg
Sascha Scheuermann, Rodenbach
Michael Derendorf/Weenen, Düsseldorf
Sven Calkster, Berlin
Christian Tütz, Stelle
Jörg Gerjets, Norden
Christian Nielsen, Hamburg
Thomas Müller, Rascheid
Peter Wetzel, Grünstadt
Christoph Oldenburg, Rödermark

Die Gewinner wurden aus allen Einsendungen zu dieser Umfrage gezogen. Die Verteilung der Programme war vollkommen zufallsbestimmt. Die Preise befinden sich sowohl auf Diskette als auch Kassette, und reichen von Spielen über Textverarbeitung bis Grafik-Software. Wir bitten um Ihr Verständnis, wenn die Preise vom Umtausch ausgeschlossen sind.



Der C 128 D im ersten Test

Auf der Systems in München konnte man zum ersten Mal den deutschen C 128 D bewundern. Wir entführten ihn zu einem ersten Test vom Commodore-Messestand.



Der C 128 im professionellen PC-Look, so präsentiert sich Commodores jüngstes Kind, der C 128 D. Das D steht für »Disk«, denn der C 128 D hat gleich ein Diskettenlaufwerk eingebaut. Einen Spitznamen hat er auch schon, denn manche Leute interpretieren das »D« als Abkürzung für Diesel ...

Keine Angst, der C 128 D läuft mit der ganz normalen Netzspannung von 220 Volt. Und hat man den C 128 D an eine Steckdose angeschlossen und eingeschaltet, meldet er sich genauso wie ein echter C 128. Ist er's aber auch? Denn viele werden sich noch mit Schrecken an den »vollkompatiblen« SX 64 erinnern, der erhebliche Schwierigkeiten mit so manchem Programm hatte, das der C 64 problemlos verarbeitete. Aber keine Angst, beim C 128 D hat sich eigentlich nur das Äußere geändert. Schraubt man ihn auf, findet man die nur minimal veränderten Platinen des C 128 und der 1571 wieder. Diese sind allerdings fantastisch verpackt, wie in Bild 1 zu sehen. Commodore hat sich noch nie soviel Mühe mit der Abschirmung der einzelnen Baugruppen gegeben wie beim C 128 D. So kommt man, will man den Computer nicht komplett auseinandernehmen, gar nicht erst an die Platinen heran. Die zahlreichen Abschirmbleche zeichnen sich nämlich durch besonders gute Verschraubung mit dem Gehäuse aus. Ein kurzer Blick in die Betriebssysteme von Computer und Laufwerk erweckte den Eindruck, daß hier überhaupt keine Änderungen vorgenommen wurden. Sollte sich dies bewahrheiten, würde das heißen, daß der C 128 D nicht nur kompatibel zum, sondern identisch mit dem normalen C 128 ist. Dies beweisen auch die identischen Anschlüsse, die wir in Bild 2 und 3 dargestellt haben. Für die abgesetzte Tastatur mußte nur eine weitere Buchse hinzugefügt werden.

Aufgrund des integrierten Floppy-Laufwerks und der abgesetzten Tastatur läßt sich mit dem C 128 D hervorragend arbeiten. Die Tastatur entspricht vollkommen der des C 128, sie wurde bloß in ein eigenes Gehäuse verfrachtet. Zu Transport- und Aufräumzwecken kann die Tastatur unter dem C 128 D festgeklinkt werden (Bild 4). Der gesamte Computer läßt sich dann an einem ausklappbaren Handgriff durch die Lande tragen. Allerdings: Der C 128 D hat ja keinen eingebauten Monitor, also kann man ein funktionsfähiges System gar nicht mal so einfach transportieren. In einer Hand den Computer, in der anderen den Monitor — diesen Zustand machen nur Bodybuilder längere Zeit mit.

Integriertes Diskettenlaufwerk

Aber das Hauptargument für den C 128 D ist ja nicht sein Handgriff. Denn wer das Gerät auf dem

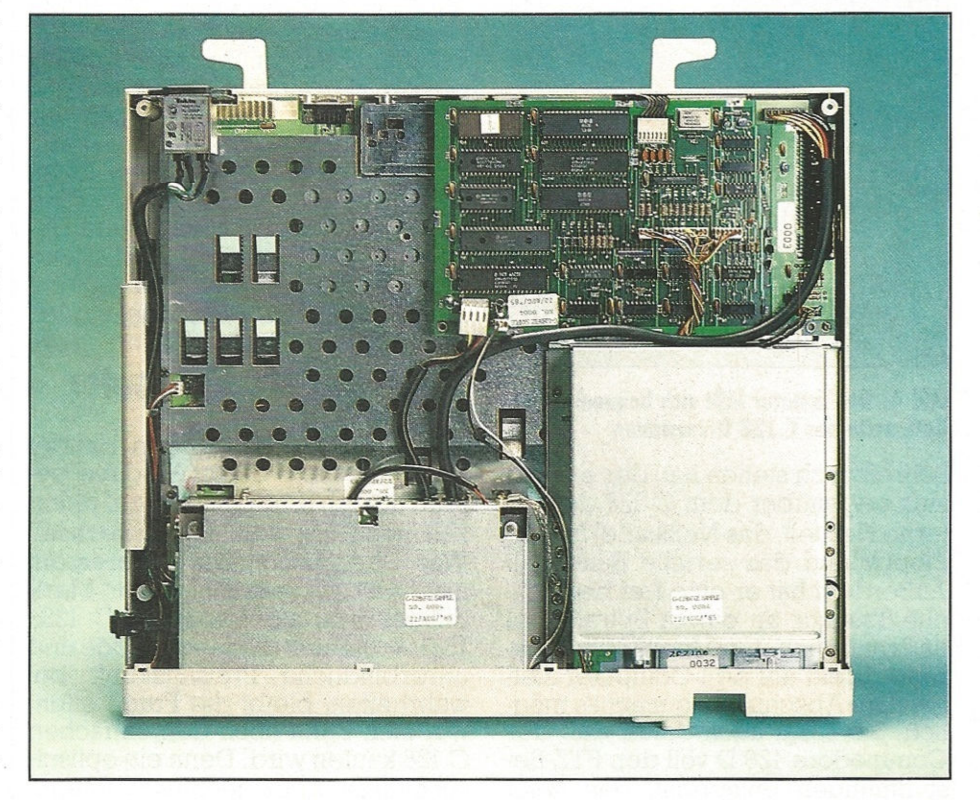


Bild 1. Die Abschirmung des C 128 D ist fast zu gut gelungen, denn ein Service-Techniker muß ein paar Dutzend Schrauben lösen, bis er die Platinen endlich sieht

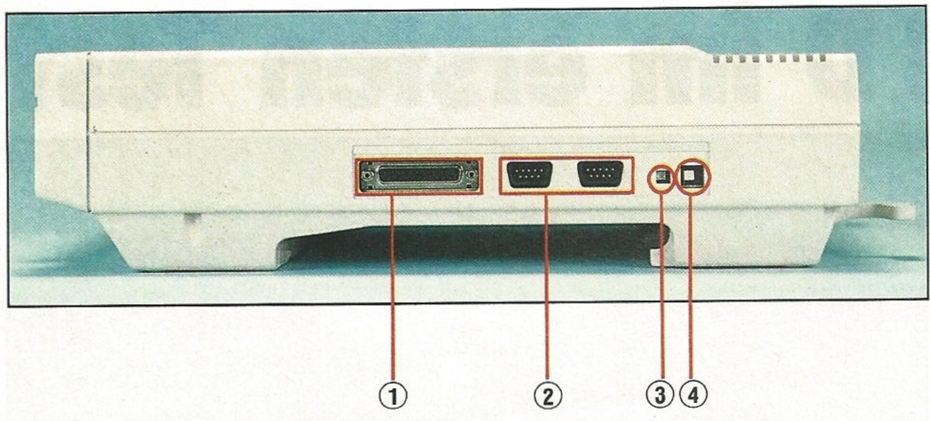


Bild 2. Die Anschlüsse des C 128 an der linken Seite: 1-Tastatur; 2-Joystickports; 3-Resettaste; 4-Netzschalter

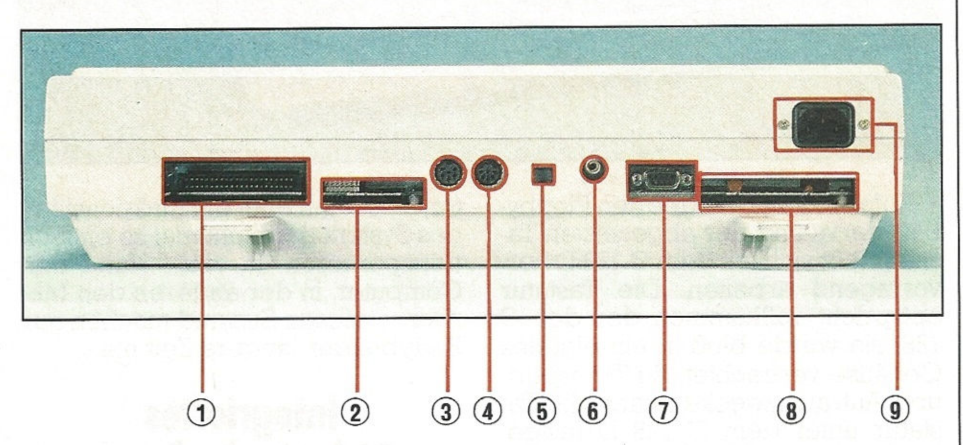


Bild 3. Die Anschlüsse des C 128 an der Rückseite: 1-Expansion-Port; 2-Datasettenanschluß; 3-Serieller Bus; 4-Videoausgang; 5-Kanalwahl (nicht bei deutschen Versionen); 6-Antennenausgang; 7-RGB-Anschluß; 8-User-Port; 9-Netzkabel



Bild 4. Die Tastatur läßt sich bequem an der Unterseite des C 128 D verstauen

Schreibtisch stehen hat, der erspart sich gegenüber dem C 128 das externe Netzteil, das Netzkabel für die Floppy und das serielle Buskabel. Außerdem hat er eine frei bewegliche Tastatur an einem Spiralkabel und man kann den Monitor platzsparend direkt auf den Computer stellen (Die Abschirmung macht's möglich. Sie sorgt auch dafür, daß der Commodore 128 D voll den FTZ-Bestimmungen entspricht, das heißt keine Störstrahlung in die Umgebung abgibt). Ein richtig professioneller Arbeitsplatz, der sich nur durch sein Innenleben von einem Commodore PC 10 oder einem IBM-PC unterscheidet. Denn der C 128 D ist wohl der einzige Computer, der sich im Büro gleichermaßen für Textverarbeitung und Datenverwaltung (»Wordstar« und »dBase II« im CP/M-Modus), Programmierung (Basic 7.0 im C 128-Modus) und Unterhaltung (»Impossible Mission«, »Summer Games« und über 1000 andere Spiele im C 64-Modus) eignet.

Kompakter Computer

Der C 128 D soll in den Handel kommen, wenn diese Ausgabe gerade an den Kiosken liegt, auf jeden Fall noch vor dem Jahreswechsel. Was ihn besonders interessant macht: Er soll etwa einhundert Mark billiger sein als das Team C 128/1571. Sollte sich diese vorläufige und unverbindliche Preisauskunft bewahrheiten, bleibt die Frage offen, wer sich dann noch den einfachen C 128 kaufen wird. Denn ein optisch schöneres aber technisch identisches Gerät für weniger Geld — wer würde da nicht zuschlagen? (bs)

Die C 128-Besitzer haben lange auf ein passendes Diskettenlaufwerk warten müssen. Jetzt sind sogar gleich zwei verschiedene lieferbar. Wir haben beide getestet und verglichen.

Bis vor kurzem mußten C 128-Besitzer noch mit der guten alten 1541 arbeiten, wollten sie Daten und Programme abspeichern und nicht auf die Datasette zurückgreifen. Die Vorteile, die der C 128 mit einem entsprechenden Laufwerk bietet, können aber mit der 1541 nicht ausgeschöpft werden. Im Klartext heißt das: quälend langsame Diskettenzugriffe, kein Zugriff auf Standard-CP/M-Formate, geringe Speicherkapazität und so weiter. Verständlicherweise warteten alle C 128-Besitzer also auf die angekündigte 1571. Doch Commodore hatte produktionstechnische Probleme mit dem neuen Gehäuse, die die Lieferfähigkeit der 1571 stark einschränkten. So wurde als Übergangslösung die 1570 konzipiert, ein Zwischending aus 1541 und 1571. Inzwischen stehen beide Laufwerke, 1570 und 1571, in den Geschäften und können von jedermann gekauft werden. Die Frage lautet bloß: Welches von den beiden? Bevor wir diese Frage beantworten, wollen wir erst einmal die gemeinsamen und die unterschiedlichen Daten der Laufwerke vorstellen.

Eines der wichtigsten Verkaufsargumente für die 1570/71-Laufwerke ist ihr stark beschleunigter Busbetrieb. Zusammen mit dem C 128 sollten Geschwindigkeiten erreicht werden, von denen ein C 64 mit 1541 ohne Beschleuniger bisher nur träumen konnte. Wer sich allerdings nun eine parallele Datenübertragung oder ähnliches vorgestellt hatte, wird enttäuscht. Der normale serielle Bus des C 64 wurde nur leicht umgebaut. Eine bisher unbenutzte Leitung wird jetzt zur Synchronisation der schnellen Datenübertragung genutzt.

Zieht man Vergleiche zur 1541, so ist man in den ersten Augenblicken angenehm überrascht: Ein Testprogramm von 200 Blöcken auf der Diskette wird vom C 64 und der 1541 in etwas über zwei Minuten geladen. Die 1570/71 schafft im C 128-Modus die 200 Blöcke in unter 14 Sekunden! Voraussetzung ist allerdings, daß



Die neuen Laufwerke Commodore-1570/1571



auch im C 128-Modus gespeichert wurde. Bei vom C 64 gespeicherten Programmen dieser Größenordnung werden 25 Sekunden benötigt, die Zeit, die auch von SpeedDos, TurboAccess oder Hypra-Load erreicht wird. Beim Speichern eben dieser 200 Blöcke erweist sich die 1570/71 als Langweiler. Sie benötigt 1 Minute 52 Sekunden gegenüber 2 Minuten 25 Sekunden bei der 1541 und etwa 1 Minute 40 Sekunden bei den hardwaremäßigen Beschleunigern. Andere Disketten-Zugriffe, wie relative und sequentielle Dateien, werden um Faktoren von durchschnittlich 2 bis 6 beschleunigt, je nach Anwendung. Die Kommandos »Scratch« und »Validate« sind nur unwesentlich schneller als bei der 1541. Bei der 1571 kann ein Validate allerdings doppelt so lange dauern, weil ja die doppelte Datenmenge bearbeitet werden muß. Das Formatieren schließlich ist auch recht schnell: Die 1570 braucht zirka 23 Sekunden, die 1571 die doppelte Zeit, weil sie ja auch doppelseitig formatiert.

Ein wenig enttäuscht hat uns der CP/M-Modus. Programme wie »Wordstar« oder »Multiplan« sind zwar in knapp einer Minute geladen. Doch da diese Programme sehr oft auf die Diskette zugreifen, sind längere Wartezeiten vorprogrammiert. Andere CP/M-Systeme arbeiten mit einem parallelen Bus, der sehr viel schneller als der serielle Commodore-Bus ist.

Natürlich haben wir die Laufwerke auch mal von innen genauer betrachtet. In den Bildern 1 und 2 können Sie die Innereien der beiden Laufwerke sehen.

Die Hardware

Auffällig ist, daß in beiden Laufwerken exakt dieselbe Platine vorzufinden ist. Die fertige 1571-Platine wurde dann auch nur durch ein leicht geändertes DOS und einige minimale Veränderungen auf der Platine für die 1570 angepaßt.

Auf der Platine befinden sich alle Chips, die man von der 1541 her kennt, und ein paar neue, die wohl für die CP/M-Modi verantwortlich sind. Die genaue Bedeutung dieser Bausteine, insbesondere des großen schwarzen Hybridbausteins (auf der 1571-Platine an der Rückseite zu finden), haben wir aber noch nicht völlig entschlüsseln können. Genauso geheimnisumwittert ist der CP/M-Controller-Chip. Hier tut sich für viele Hardware-Freaks ein weites Betätigungsfeld auf.

Kommen wir nun zur Mechanik, ein bei Floppy-Laufwerken sehr wichtiger Aspekt. Hier zeigen sich die ersten großen Unterschiede zwischen den beiden Modellen. Während die 1570 eine leicht verbesserte Version der 1541-Mechanik verwendet, ist die der 1571 völlig neu entwickelt worden. Die Verbesserungen in der 1570 beschränken sich auf zwei Lichtschranken. Die eine fragt das Indexloch ab und wird bei manchen CP/M-Formaten verwendet. Die zweite Lichtschranke stoppt den Schreib/Lese-Kopf eine Halbspur vor dem Anschlag am Steppermotor und verhindert somit das rhythmische Klappern beim Formatieren und bei Lesefehlern. Diese beiden Lichtschranken sind ebenfalls in der 1571 vorhanden. Die Mechanik der 1571 macht allerdings einen professionelleren und stabileren Eindruck als die der 1570. Wichtigstes Unterscheidungsmerkmal: Die 1571 hat einen doppelten Schreib/Lese-Kopf, mit dem auf beide Seiten der eingelegten Diskette zugegriffen werden kann. Sehr vorteilhaft ist uns die völlige thermische Entkopplung von Mechanik, Netzteil und Elektronik aufgefallen. So dürften bei der 1541 die bekannten Überhitzungsprobleme, die zu verstellten Köpfen und defekten ICs führten, beseitigt sein. Einzige Minuspunkte der 1571 sind der zerbrechlich wirkende Knebelverschluß und der von schräg oben nur sehr schwer erkennbare Diskettenschacht.

Das DOS

Die Commodore-Laufwerke waren schon immer »intelligent«, das heißt, sie haben ein eigenes Betriebssystem und beanspruchen praktisch keinen Speicherplatz im Computer. Das hat sich bei den beiden neuen Laufwerken nicht geändert. Sie haben das Commodore-DOS 3.0 eingebaut. DOS 3.0 unterscheidet sich von den DOS 2.X-Versionen (1541, SFD 1001, etc.) in mehreren Punkten. Es unterstützt neben den Commodore-GCR-Formaten auch CP/M-MFM-Formate, bietet den schnellen Busbetrieb und hat zu guter Letzt ein neues Kommando. Mit dem Kommando »U0« kann eine ganze Reihe von Funktionen gesteuert werden. Leider hat Commodore wieder die »Glanzlei-

stung« vollbracht, das neue Kommando im deutschsprachigen Handbuch nur in einer kleinen Tabelle und dann auch noch unvollständig zu beschreiben. In Tabelle 1 können Sie alle Funktionen finden, die mit »U0« angesprochen werden können, und deren Bedeutung entweder aus dem englischen Handbuch entnommen beziehungsweise von unserem Floppyexperten Karsten Schramm herausgeknobelt wurde. Manche Funktionen sind noch nicht völlig geklärt, insbesondere die, die sich auf die MFM-Formate beziehen.

Bis auf das »U0«-Kommando ist das DOS mehr oder minder identisch mit dem der 1541 (DOS 2.6). Natürlich wurden noch einige kleinere Veränderungen vorgenommen, um das DOS an die höhere Speicherkapazität anzupassen.

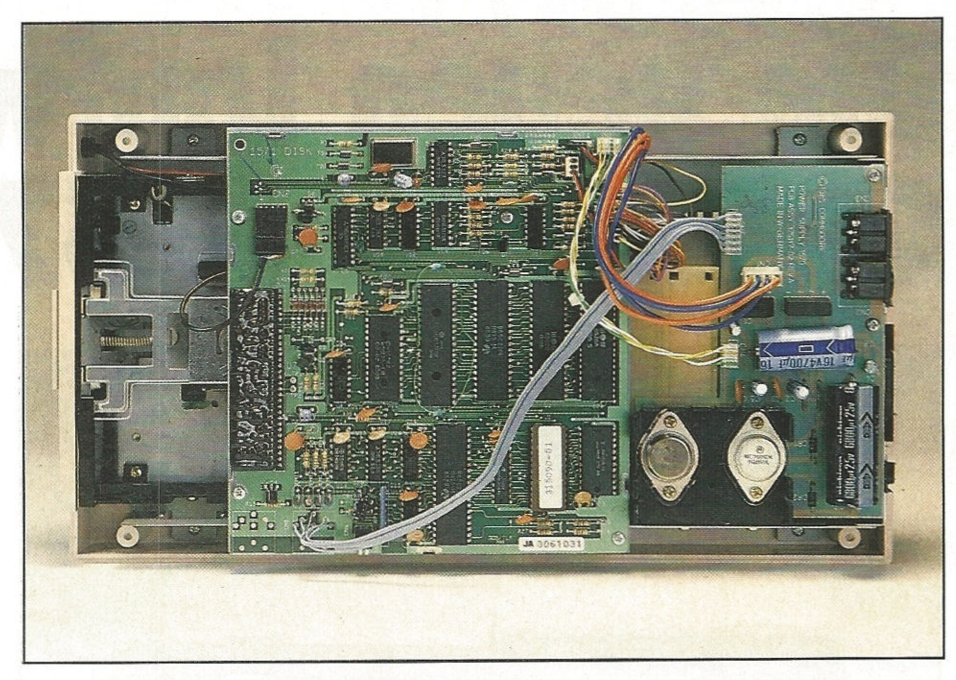


Bild 1. Das Innenleben der 1570 präsentiert sich als Mischung aus der 1541-Mechanik und 1571-Elektronik. Die Speicherkapazität beträgt bis zu 200 KByte.

Die Formate

Die 1570/71-Laufwerke gehören zu den flexibelsten überhaupt, denn sie können viele verschiedene Formate lesen und schreiben. Dazu gehört zuerst einmal das Format der 1541. Jede von einer 1541 beschriebenen Diskette kann ganz normal weiterverwaltet werden. Die 1571 kennt sogar einen speziellen 1541-Modus, in dem sie fast identisch zur 1541 wird. Selbst das Formatieren läuft dann wieder ganz gemächlich ab. Dann gibt es natürlich das neue 1571-Format, das dem der 1541 weitgehend entspricht. Bei zweiseitiger Benutzung der Disketten werden BAM und Directory allerdings etwas anders organisiert. Und als letztes stehen mehrere MFM-Formate zur Verfügung. MFM ist die Abkürzung für Modified Frequency Modulation (Modifizierte Frequenz Modulation) und steht für ein bestimmtes Aufzeichnungsverfahren einzelner Bits auf einer Diskette. MFM-Formate werden von vielen CP/M-Computern genutzt. Die 1570/71 ist in der Lage, MFM-Formate zu lesen und zu schreiben, allerdings nur blockorientiert. Das heißt, daß die notwendige Intelligenz zur Verwaltung von Files, Directories und ähnlichem wieder beim Computer liegt. Ein Schritt zurück also? Ja und doch wieder nein, da die CP/M-Formate so und so vom Computer aus verwaltet werden. Und das ist kein Einfall von Commodore, sondern schon seit Jahren bei allen CP/M-Computern gang und gäbe. Erst dadurch sind CP/M-Disketten zwischen verschiedenen Computer-Systemen austauschbar.

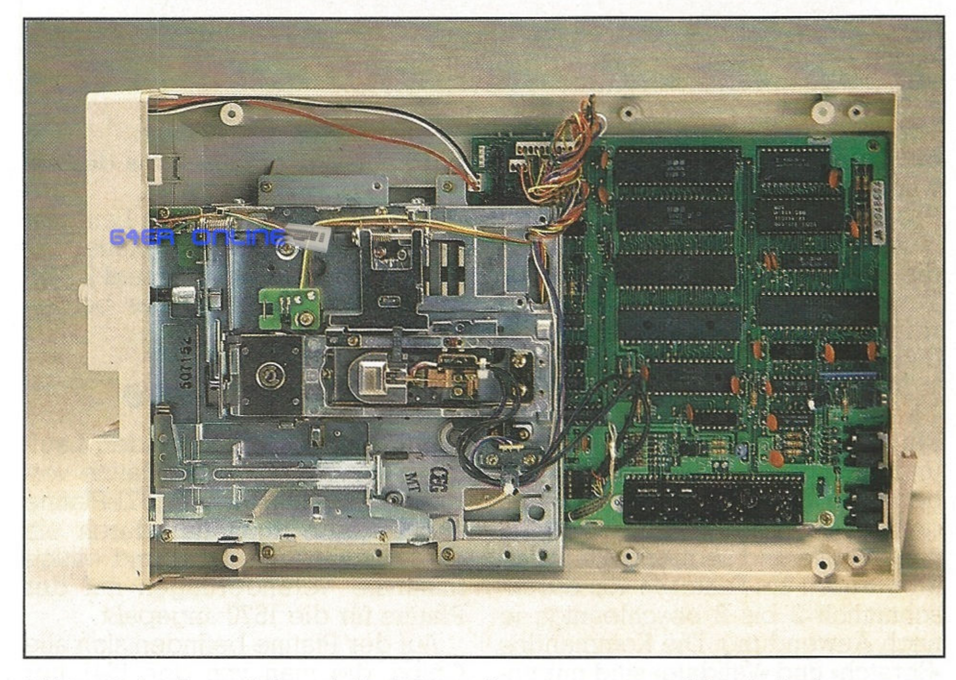


Bild 2. Mechanik wie Elektronik der 1571 machen einen guten stabilen Eindruck. Das abgeschirmte Netzteil über der Platine wurde für das Foto entfernt.

Dadurch, daß die 1571 verschiedene MFM-Formate bearbeiten kann, kann man von allen gängigen CP/M-Computern die Disketten verwenden, angefangen beim tragbaren Osborne und Kaypro bis hin zum IBM-PC oder PC 10. Die Diskette muß nur unter dem Betriebssystem CP/M (und zwar CP/M 86 bei MS-DOS-Computern!) angelegt worden sein, damit das Gespann C 128/1570/71 an die Daten herankommt.

Die bange Frage aller Auf- und Umsteiger vom C 64 auf den C 128 lautet: Ist das Laufwerk so konzipiert, daß meine 1541-Programme auch auf der 1571 laufen? Wie sieht es mit floppy-spezifischen Programmen wie Kopierprogrammen und Kopierschutzprogrammen aus? Die potentiellen Käufer der 1570/71 können relativ beruhigt sein. Commodore hat so viel wie möglich unternommen, um die Kompatibilität zu wahren. Die meisten Programme, die in das DOS eingreifen, funktionieren noch, angefangen bei Hypra-Load bis zum Turbo-Nibbler. Einige Programme laufen allerdings nicht, so zum Beispiel das Kopierprogramm Quickcopy. Probleme gab es außerdem mit so manchem neue-



ren Kopierschutz. Denn gerade die neuen Schutzmechanismen verwenden sehr unübliche Befehlsfolgen, die bei den teilweise veränderten DOS-Routinen in der 1570/71 zum Absturz führen.

Fazit

Für den ernsthaften C 128-Anwender ist der Kauf einer der beiden neuen Laufwerke gar keine Frage. Die Vorteile gegenüber der 1541 sind so stark, daß diese als billigere Alternative fast nicht in Frage kommt. Schneller Busbetrieb und die MFM-Formate ermöglichen erst ein vernünftiges Arbeiten insbesondere für CP/M-Benutzer. Die Frage bleibt, welches der beiden Laufwerke besser ist. Geht man mal vom Preis/Leistungs-Verhältnis der beiden aus, ist die 1571 sicherlich attraktiver. Denn für nur 200 Mark mehr (1570 zirka 798 Mark, 1571 zirka 998 Mark) erhält man doppelte Speicherkapazität und eine erheblich bessere Mechanik. Wer sich allerdings mit der 1570 zufriedengibt, ist auch nicht schlecht bedient. Ob es das 1570-Laufwerk aber auch noch nach dem Frühjahr 1986 geben wird, steht in den Sternen. (bs)

Das u0-Kommando

u0:>S chr$(x)	— Sector Interleave einstellen (Abstand der Sektoren beim Schreiben)
u0:> chr$(x)	— Anzahl der Leseversuche festlegen
u0:> T	— ROM-Prüfsumme erstellen
u0:> M1	— schalten auf 1571-Modus
u0:> M0	— schalten auf 1541-Modus
u0:> H0	— Seite 0 anwählen (nur 1541-Modus)
u0:> H1	— Seite 1 anwählen (nur 1541-Modus)
u0:> chr$(x)	— Gerätenummer einstellen

Tabelle 1. Diese Aufzählung zeigt einige der Funktionen des u0-Kommandos bei der 1571. Diese Kommandos sind bis auf »H0;H1« auch bei der 1570 verwendbar. Weitere Funktionen, die über u0 gesteuert werden, sind die MFM-Formate und der besonders schnelle »Burst«-Modus (wird beim Laden von Programmen schon ausgenutzt). Diese Funktionen befinden sich nicht in dieser Tabelle.

(Karsten Schramm/bs)

Gestatten: Wordstar

Professionelle Textverarbeitung unter CP/M — der C 128 macht's möglich.

Mit Wordstar steht dem C 128-Besitzer ein Textverarbeitungsprogramm aus dem Bereich der echten Personal Computer zur Verfügung. Um es vorweg zu sagen: Wordstar ist ganz sicher nicht die schnellste Textverarbeitung, die für einen Computer wie den C 128 denkbar ist, aber die Leistungsfähigkeit dieses professionellen Programms macht das Geschwindigkeitsmanko mehr als wieder wett.

Doch beginnen wir ganz von vorn: Wordstar für den C 128 wird auf zwei Disketten geliefert. Eine Diskette (Nummer 2) enthält das uninstallierte Wordstar, eine Art Rohversion, die mittels eines ebenfalls mitgelieferten speziellen »Install«-Programms an einen speziellen Computer angepaßt werden kann. Damit braucht man sich bei einem späteren Wechsel des Computers sein Textverarbeitungsprogramm nicht ein zweites Mal zu kaufen: Mit Hilfe des Install-Programms und des sehr ausführlichen Handbuches kann man Wordstar an jeden CP/M-fähigen Computer anpassen. Die zweite Diskette enthält Wordstar schon fix und fertig installiert und an den C 128 angepaßt — und das gleich in zwei Versionen: Das File WSCBM.COM ist eine zu Commodore-Druckern kompatible Wordstar-Version.

Centronics-Schnittstelle eingebaut

Die zweite Wordstar-Version, WSPAR.COM, unterscheidet sich von der ersten dadurch, daß hier softwaremäßig eine Centronics-Schnittstelle über den User-Port realisiert wurde.

Wordstar ist nicht kopiergeschützt und gestattet dem Käufer das Anfertigen von Sicherheitskopien für den persönlichen Gebrauch.

Nach dem Laden meldet sich Wordstar mit einem übersichtlichen Startmenü. Da es die wichtigsten Befehle klar und eindeutig auflistet, erspart dieses Menü den Griff zum Handbuch.

So ist im Startmenü leicht der Befehl zum Dateieröffnen zu finden: Nur den Buchstaben »D« (wie »Datei«) eingeben. Nach Eingabe eines Namens kann man schon mit der Abfassung seines Textes beginnen. Wenn der Name schon auf Diskette existiert, wird der entsprechende Text geladen.

Die obere Hälfte des Bildschirms enthält dabei das sogenannte »Haupt-Kommando-Menü«, das dem Benutzer ständig die wichtigsten Befehle präsentiert. Ein Untermenü von einer Zeile hält alle übrigen Wordstar-Funktionen bereit. Es gibt Funktionen zum zeichen-, wort-, zeilen- oder abschnittsweisen Weitergehen im Text, man kann wahlweise durch den Text »blättern« oder rollen, Zeichen, Worte oder Zeilen löschen. Auch Blockoperationen stehen zur Verfügung: Man kann beliebige Textabschnitte als Blöcke markieren, diese dann mit einfachen Befehlen löschen, kopieren, verschieben, speichern oder als Textbausteine definieren. Automatisches Suchen von Zeichenfolgen im gesamten Text oder auch nur in bestimmten Abschnitten ist sehr einfach möglich. Auch Suchen mit automatischem Ersetzen der gefundenen Zeichenfolge durch eine andere ist möglich; wahlweise mit oder ohne Rückfrage beim Benutzer.

Der einmal eingegebene Text kann durch nachträgliche Formatierung in eine beliebige äußere Form gebracht werden. Die eingebaute, abschaltbare »Trennhilfe« veranlaßt bei langen Wörtern eine Unterbrechung eines eingeleiteten Formatiervorgangs und macht an entsprechender Stelle einen Trennvorschlag.

Fußnoten — kein Problem

Spezielle »Punkt-Kommandos« steuern die Ausgabe des Textes in einem bestimmten Druckbild. Das ».FO«-Kommando bewirkt zum Beispiel, daß Sie eine Fußnote, die sonst mühsam unterhalb des Textblockes angefügt werden mußte, nun problemlos im Gesamttext mitschreiben können: Die Fußnote wird dennoch sauber ans Seitenende gesetzt. Andere Punktbefehle steuern Seitennumerierung, Hoch- und Tiefstellen von Zeichen und legen die Randbegrenzungen für den Ausdruck fest. Außerdem können Kopfzeilen definiert werden, die zu An-

fang jeder neuen Seite immer wieder gedruckt werden.

Eine weitere sehr interessante Anwendung ist die Definition von »Textvariablen«, die erst beim Ausdruck durch entsprechende Daten — zum Beispiel aus einer Adreßdatei — ersetzt werden.

Auf diesem Grundgedanken basiert nun ein ganzes Programm, nämlich »Mailmerge«, das einen wesentlichen Teil der in einem Büro anfallenden Schreibarbeiten erspart. Mailmerge erlaubt es, auf einfache Weise auf Adreßdateien oder Textbaustein-Dateien zuzugreifen und erweitert somit die Anwendungsgebiete von Wordstar ganz enorm. Der Wordstar-Besitzer kann diese Funktionen von Anfang an in Anspruch nehmen: Mailmerge befindet sich ganz einfach auf der Wordstar-Diskette und ist im Handbuch ausführlich dokumentiert.

Fazit

Wordstar ist ganz ohne Zweifel eines der leistungsfähigsten Textverarbeitungs-Programme, die je für einen Heimcomputer zu haben waren. Fähigkeiten wie die automatische Verwaltung von Fußnoten oder das Verarbeiten von Textbausteinen heben das Programm über das Niveau so mancher anderen Textverarbeitung hinaus. Das sehr umfangreiche Handbuch beantwortet alle Fragen, die im Zusammenhang mit der Benutzung oder der Installation von Wordstar auftreten können. Einziger Schwachpunkt ist die doch relativ langsame Textausgabe auf dem Bildschirm, die aber in erster Linie vom sehr umständlich geschriebenen Commodore-BIOS für das CP/M-System abhängt.

Ein großer Vorteil von Wordstar liegt aber in der weitgehenden Unabhängigkeit von einem bestimmten Computersystem. Ein und dasselbe Programm, einmal gekauft, bleibt auch bei Anschaffung eines neuen Computers noch aktuell — sofern der Computer CP/M-fähig ist. Und selbst beim »Aufstieg« in die Höhen der 16-Bit-Welt der IBM-kompatiblen Computer können wenigstens die alten Texte und Dateien weiterverwendet werden. Die 199 Mark für dieses Textprogramm sind also auch langfristig gesehen gut angelegt — Wordstar kann eine Anschaffung fürs Leben sein.

(Eva-Maria Hierlmeier/ev)

Info: Wordstar ist ein eingetragenes Warenzeichen von MicroPro International, USA. Vertrieb in Deutschland: Markt & Technik Verlag, Hans-Pinsel-Str. 2, 8013 Haar bei München. Wordstar 3.0 für den C 128 kostet 199 Mark

dBase II — die professionelle Datenbank

Vergessen Sie alle Dateiverwaltungs-Programme für den C 64/C 128, die Sie bis jetzt kennen: Hier ist ein echtes Datenbanksystem mit eigener Programmiersprache zu einem unverschämt günstigen Preis.

Bis vor kurzer Zeit gab es noch eine strikte Trennung zwischen Heimcomputern für Spiel und Unterhaltung und den Personal Computern (PC) für professionelle Anwendungen im geschäftlichen Bereich. Der Commodore 128 und die Schneider-Computer haben diese Grenzziehung aufgeweicht: PC-Software, entwickelt für den professionellen Anwender, steht jetzt auch für den Heimbereich zur Verfügung. Oder andersherum gesehen: Professionelle Anwendungen sind jetzt auch mit den neuen Heimcomputern der 128 KByte-Klasse möglich geworden — und das zu einem Preis, der für PC-Verhältnisse direkt anstößig niedrig ist: Für je 199 Mark erhält der C 128-Besitzer mit Wordstar, dBase II und Multiplan drei für den PC-Bereich entwickelte Programme, die mit praktisch gleicher Ausstattung und Leistung für einen PC gleich das Vielfache kosten.

Das leistet eine Datenbank

Die Aufgabe einer Datenbank ist es, Informationen in geordneter Form zu verwalten und für den Benutzer bereitzustellen. Bei echten Datenbanken erschöpft sich das Bereitstellen von Daten nicht in der (bescheidenen) Fähigkeit, nach bestimmten Begriffen suchen zu können und einen Datensatz als Text auszugeben. Es sind vielmehr sehr komplexe Verknüpfungen einzelner Daten ebenso möglich, wie die Anwendung bestimmter Funktionen auf beliebig große Teile einer Datenbank in einem logischen Arbeitsschritt.

Diese kurze und sicherlich nicht ganz vollständige Beschreibung zeigt schon auf, daß etwa ein menügesteuertes Programm, bei dem der Anwender nur unter ganz bestimmten, vorgegebenen Funktionen wählen kann, niemals die Leistung einer Datenbank erreichen kann.

Dementsprechend ist dBase II in Form einer eigenen Programmiersprache erstellt worden.

So arbeitet man mit dBase II

Über einfache Befehle können sehr komplexe Datenstrukturen festgelegt werden. Es ist beispielsweise möglich, zwei verschiedene Tabellen zu einer einzigen neuen zu verbinden, oder einen Datenbestand nach neuen Gesichtspunkten zu reorganisieren. Sie brauchen aber keine Angst vor komplizierter Programmierung zu haben. Einfache Anwendungen, wie beispielsweise eine Adreßverwaltung oder eine Schallplattendatei, sind für dBase II-Verhältnisse so primitiv, daß eine besondere Programmierung gar nicht nötig ist.

dBase unterstützt den Benutzer nämlich, indem es alle relevanten Daten im Dialog abfragt. Betrachten wir als Beispiel das Anlegen einer Adreßverwaltung mit Datenfeldern für Name, Vorname, Straße, Wohnort. Es könnte sich folgender Dialog ergeben (Eingaben des Anwenders in Kleinschrift):



```
create
BITTE DATEINAMEN EINGEBEN: adress
SATZSTRUKTUR FOLGENDERMASSEN
EINGEBEN:
FELD  NAME, TYP, LAENGE, DEZIMAL-
STELLEN
001   name, c, 20
002   vorname, c, 15
003   strasse, c, 25
004   wohnort, c, 25
005
DATEN JETZT EINGEBEN?
j
```

Hat man die letzte Frage mit »j« beantwortet, dann erscheint die Meldung »SATZNUMMER 00001« und darunter eine Eingabemaske mit den vier Feldern für Name, Vorname, Straße und Wohnort. Alle Felder wurden als nichtnumerisch (»c« wie »Character«) definiert. Die Länge der einzelnen Eingabefelder ist durch die Längenangabe bei der Satzstruktur vorgegeben.

Man kann jetzt Datensatz um Datensatz eingeben; es ist aber auch möglich, Datensätze oder Teile davon aus anderen, bereits bestehenden Dateien zu übernehmen.

Um die spätere Suche nach bestimmten Einträgen zu beschleunigen, können beliebig viele Datenfelder indiziert werden (Befehl »INDEX ON...«).

Selbstverständlich können Daten verändert oder neu hinzugefügt werden und das nicht nur direkt »per Hand«, sondern es können andere Dateien herangezogen werden, um automatisch Daten zu ändern, zu löschen, anzufügen, zu mischen oder zu sortieren. Sofern numerische Datenfelder vorgesehen sind, können auch Rechenoperationen angewendet werden. Mit einem dBase-Kommando ist es beispielsweise möglich, irgendwelche in einer Datei festgehaltenen Kosten zusammenzuzählen, die Ergebnisse in anderen Datenfeldern festzuhalten oder auszugeben.

Ein besonders benutzerfreundliches Feature von dBase II ist der eingebaute Report-Generator. Möchte man gerne eine Übersicht über irgendwelche Datenbestände, entweder allgemein oder nach bestimmten Gesichtspunkten, dann kann man im Dialog mit dBase, ohne Programmierkenntnisse, ein Report-Programm erstellen, das gespeichert werden kann und bei jedem Aufruf eine aktuelle Übersicht über den Datenbestand ausgibt.

Doch damit sind die Möglichkeiten von dBase II noch lange nicht erschöpft. Wer etwas von Programmierung versteht, kann sich sehr leistungsfähige Datenbanksysteme mit eigenem Bildschirmlayout und eigener Benutzerführung selbst zusammenstellen. dBase II enthält nämlich neben sehr leistungsfähigen Datenbankbefehlen auch alle wichtigen Elemente höherer Programmiersprachen: IF-THEN-ELSE-Abfragen, DO-WHILE-Schleifen, Rechenoperationen und die Möglichkeit, Benutzereingaben in Variablen abzulegen und weiterzuverwenden. Insgesamt kennt dBase über hundert Befehle, Funktionen und Operatoren und ist damit ein leistungsfähiges System für eigene Entwicklungen. Andererseits werden alle wichtigen Funktionen über Dialoge mit dem Benutzer ausgeführt, so daß dBase II nicht unbedingt Programmierkenntnisse voraussetzt. Und sollte man wirklich einmal nicht mehr weiterwissen, dann hilft entweder die eingebaute HELP-Funktion oder das sehr ausführliche Handbuch weiter.

Power auf Dauer

Insgesamt ist dBase II also ein sehr leistungsfähiges System, bei dem es kaum denkbar erscheint, daß der Anwender jemals an irgendwelche Leistungsgrenzen stößt. Wenn es Grenzen gibt, dann sind es nur die durch das verwendete Computersystem gesetzten. Und auch hier liegt ein wichtiges Plus für dBase II: Falls man sich doch irgendeines schönen Tages für einen größeren Computer entschließt, war die Investition von 199 Mark nicht verloren: Handelt es sich um einen CP/M-Computer, dann kann man dBase II ganz einfach weiterverwenden. Denn wie alle wirklich professionelle Software kommt auch dBase II ohne Kopierschutz aus. Dafür wird ein Installationsprogramm mitgeliefert, das es ermöglicht, dBase auf jeden beliebigen CP/M-Computer zu übertragen. Aber auch beim Aufstieg in die 16-Bit-Klasse der echten Personal Computer können zumindest die Datenbestände weiterverwendet werden, sofern man sich dBase II in der Version für den 8086/8088-Prozessor unter MS-DOS besorgt. Viele Gründe also, die nicht nur für dBase II sprechen, sondern für CP/M überhaupt, sofern es um professionelle Software geht, die keine Eintagsfliege sein soll. (ev)

Info: dBase II ist ein eingetragenes Warenzeichen von Ashton Tate, USA. Vertrieb in Deutschland: Markt & Technik Verlag AG, Hans-Pinsel-Str. 2, 8013 Haar bei München. dBase II für den C 128 kostet 199 Mark

Aktuell

EPROM-King für C 128

Den seit einiger Zeit für den C 64 erhältlichen »EPROM-King« (Test Ausgabe 8/84) gibt es nun auch in einer Software-Version für den C 128. Neben einer umfangreichen Liste der programmierbaren Typen (2516 bis 2564, 2716 bis 27256, 27512 über Adapter, 2816, 9664A) zeichnet sich dieses Gerät durch seine sehr komfortable Software zur Bedienung aus. Darin enthalten ist unter anderem ein leistungsfähiger Maschinensprache-Monitor. Der EPROM-King ist ferner einer der wenigen, die spezielle ROM-Typen auslesen können: 2332, 4732, 2364, 4764, 23128, 47128, 23256, 47256.

Der Vorteil der C 128-Version liegt im größeren Programmspeicher, der auch die Programmierung von 32-KByte- und 64-KByte-EPROMs in einem Durchgang zuläßt.

Info: Ing. Büro H. Kalawski, Friedrich-Ebert-Str. 41, 6108 Weiterstadt 1, Tel. 06150/2541, Preis: 252 Mark

Familien-Computercamp an der Adria

Von Juni bis September 1986 bietet »Freizeit Aktiv« EDV-Lehrgänge an. Der Veranstalter sieht die Lehrgänge für Selbständige aus Klein- und Mittelbetrieben vor, die den Urlaub mit der Familie zusammen verbringen und nebenbei ihre EDV- und Basic-Kenntnisse erweitern wollen.

Eine Ferienanlage in der Nähe von Venedig mit Hotel, Bungalows und Campingplatz, soll einen angenehmen Aufenthalt sichern. Die Kurse werden an den Commodore-Computern C 64, C 128 und PC 10 durchgeführt.

Die Lehrgänge dauern jeweils eine Woche. Der Grundkurs EDV-Einführung dauert 20 Stunden und kostet etwa 440 Mark. Der Aufbaulehrgang ist etwas kürzer: Die Kursdauer beträgt 16 Stunden. Gegeben wird eine Basic-Einführung. Die Kosten betragen etwa 240 Mark. Das Lehrangebot soll von Basic-Grundlagen bis zu kommerzieller Anwendungsprogrammierung reichen. (cg)

Info: Freizeit aktiv, Postfach 4243, 6300 Darmstadt, Tel. (06251) 76338

Grafik-Probleme auf dem C 128

Anrufe von überraschten Lesern trafen in den letzten Tagen in unserer Redaktion ein: »Euer Programm für die 80-Zeichen-Grafik (Ausgabe 12/85, Seite 78) funktioniert nicht auf meinem C 128!«. Wird die hochauflösende Grafik eingeschaltet, sieht man auf dem rechten Rand ein »Kräuseln« oder Rauschen, das sich durch keine Maßnahme beseitigen läßt. Dieses Kräuseln ist allerdings nicht programmabhängig: andere Programme, die die hochauflösende Grafik im 80-Zeichen-Modus nutzen, bringen ebenfalls diese Störung am rechten Rand auf den Schirm. Allerdings sieht man das Kräuseln nicht auf allen C 128-Geräten.

In unserer Redaktion befinden sich sowohl Geräte mit, als auch welche ohne diesen Fehler. Über die Ursachen des Kräuselns kann nur spekuliert werden, selbst Commodore kann noch keine genaue Erklärung geben. Das abgedruckte Programm ist, um es noch einmal zu betonen, fehlerfrei. (bs/ev)

Ascompiler 64

Ascompiler 64 ist ein sehr schneller Compiler. Es ist das ideale Werkzeug für alle C 64-Besitzer, denen Basic zu langsam und Maschinensprache zu kompliziert ist.

Die Bezeichnung für diesen Compiler ist eine Wortbildung aus Assembler und Compiler. Ascompiler 64 ist keiner der bekannten Compiler, mit dem Sie Ihre Basic-Programme praktisch unverändert übersetzen können und dadurch etwa die dreifache Geschwindigkeit erreichen. Nein, der Ascompiler 64 ist gleichzeitig mehr und weniger: Er übersetzt nur einen sehr stark eingeschränkten Basic-Befehlssatz mit recht vielen Restriktionen, aber er übersetzt dafür in einen derart effizienten Code, daß praktisch kaum noch ein Unterschied zu Assembler besteht. Da Ascompiler 64 wirklich reinen Maschinencode erzeugt, ist er hervorragend dazu geeignet, zeitkritische Routinen zu compilieren, um sie dann in andere Basic-Programme einzusetzen. Ein kleines Beispiel soll die Leistungsfähigkeit dieses Compilers zeigen. Geben Sie doch einmal die folgenden Basic-Zeilen in Ihren Computer ein:

```
10 let a=0
20 let a=a+1
30 if a< 60000 then 20
40 print"fertig"
```

Schauen Sie jetzt auf Ihre Uhr und tippen Sie »RUN«. Es passiert zunächst sehr lange Zeit gar nichts ... die Minuten ziehen sich dahin ... dann endlich, nach ungefähr elf Minuten kommt die Meldung »Fertig«.

Was schätzen Sie, wie lange braucht das gleiche Programm nach der (blitzschnellen) Übersetzung? Eine Minute, eine halbe Minute, etwa nur 10 Sekunden? Weit gefehlt, dieses Programm läuft in compilierter Form in knapp drei Sekunden ab! Das bedeutet eine Geschwindigkeitserhöhung um den Faktor 200, vergleichbar mit der Programmierung in Assembler.

Diese auch für einen Compiler enorme Geschwindigkeit konnte nur durch eine sehr stark optimierende Übersetzung erreicht werden. Der Compiler kennt daher leider recht viele Einschränkungen: Als Variable sind nur »A« bis »Z« erlaubt, sämtliche Rechenoperationen sind in 16-Bit-Arithmetik ausgeführt, Fließkommarechnungen und mathematische Funktionen wie SIN, COS, SQR etc. sind unmöglich. Weiter können Operationen nicht geschachtelt werden, Klammerrechnung ist nicht möglich.

Folgende Basic-Befehle können bearbeitet werden: REM, CLR, END, GOTO, GOSUB, RETURN, SYS, LET (muß verwendet werden!), PRINT, POKE, PEEK, IF...THEN und STOP.

Ascompiler 64 ist nicht zur Übersetzung normaler Basic-Programme geeignet. Vielmehr ist Ascompiler 64 der ideale Ersatz zur Programmierung in Maschinensprache. Die übersetzten Programme können in beliebige Speicherbereiche gelegt und mit SYS aufgerufen werden. Insbesondere schnelle Spielprogramme lassen sich mit dem Ascompiler 64 sehr bequem schreiben.

Der Ascompiler 64 ist keine Eintagsfliege — wir werden laufend neue Programme, Tips & Tricks und Erweiterungen dafür vorstellen, sofern Sie, unsere Leser, uns bei diesem Projekt unterstützen.

Also frisch ans Werk! Tippen Sie den Ascompiler 64 ab und schreiben Sie anschließend die schnellsten Basic-Programme Ihres Lebens! (ev)

Gesucht: Die besten Programme für den Ascompiler 64

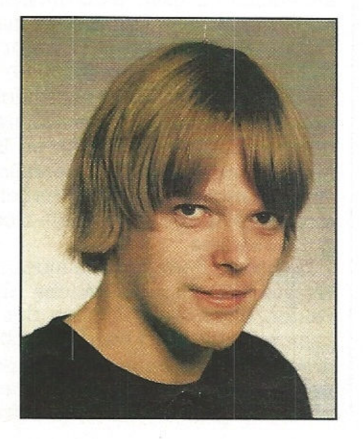
Der Ascompiler 64 ist ideal für eine Vielzahl von zeitkritischen Anwendungen. Grafik, Dienstprogramme, vor allem aber auch komplette Action-Spiele sind seine Domäne. Hier ist die Chance für alle Basic-Programmierer: Wir suchen die interessantesten Programme und Routinen, die mit dem Ascompiler 64 entwickelt wurden.

Eingeschickt werden kann jede Art von Programmen, egal ob kurz oder lang. Einzige Bedingung: Das Programm muß vollständig im Ascompiler-Format geschrieben sein, Mischprogramme aus normalem und compiliertem Basic sind nicht erlaubt.

Damit es auch richtig interessant wird, erhalten die besten Einsendungen das Prädikat »64'er Top-Listing« und eine Top-Prämie von 300, 500 oder gar 1000 Mark, je nach Qualität und Länge. Alle Programme für den Ascompiler 64, die bis zum 1.2.1986 bei uns eingehen, nehmen an diesem Wettbewerb teil. Aber auch wenn Sie nicht unter den Gewinnern der Top-Prämie sein sollten, die Teilnahme lohnt sich trotzdem. Denn selbstverständlich hat jedes Ascompiler-Listing nicht nur die Chance, auch außerhalb des Wettbewerbs für ein vernünftiges Honorar abgedruckt zu werden, sondern nimmt automatisch auch an unserem Wettbewerb zum »Listing des Monats« teil.

Schicken Sie Ihre Programme auf Diskette oder Kassette (nur das Basic-Programm, compilieren tun wir's selber) mit einer ausreichenden Beschreibung an:

Markt & Technik Verlag AG
Redaktion 64'er
Stichwort: Ascompiler-Wettbewerb
Hans-Pinsel-Str. 2
8013 Haar bei München

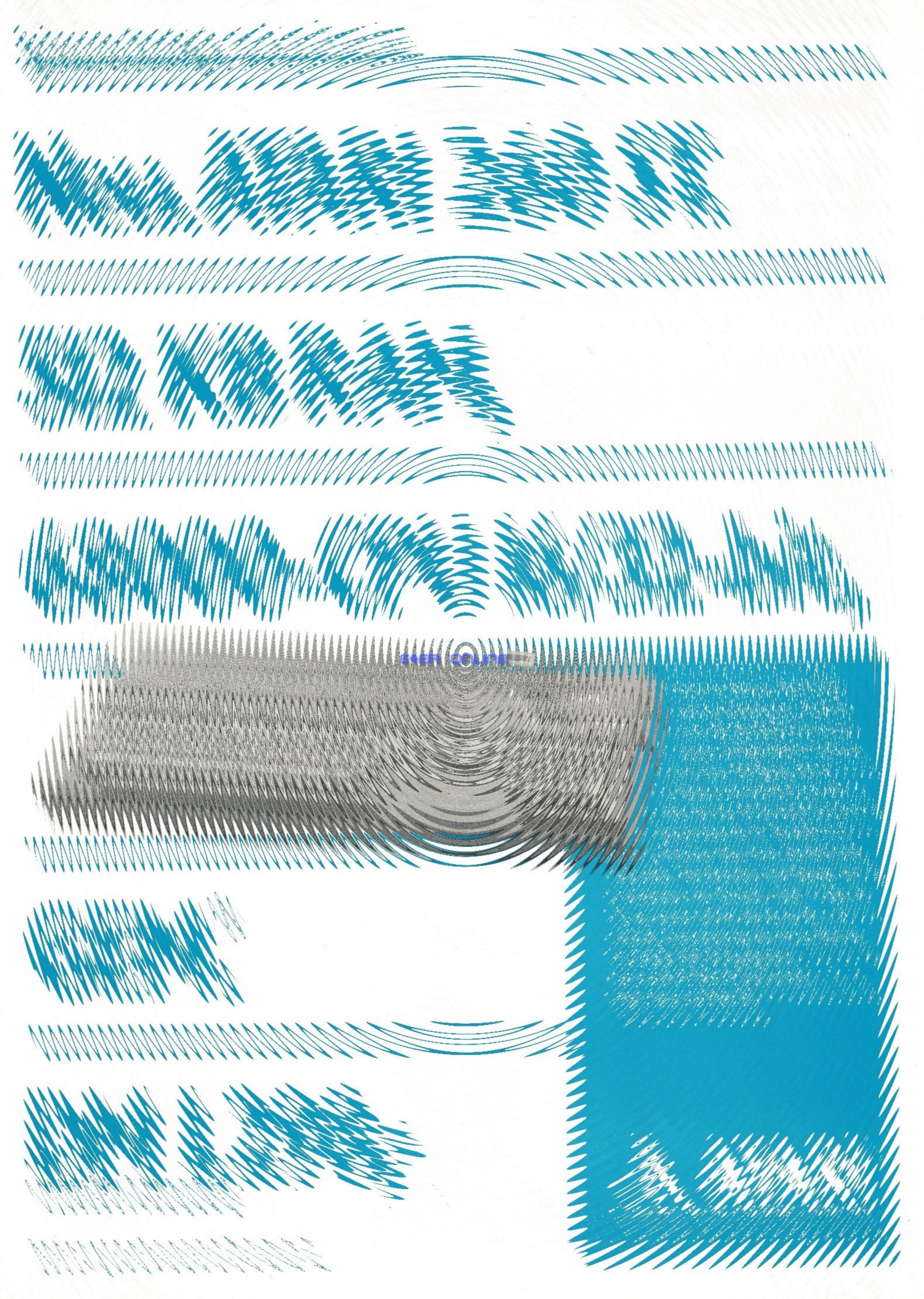


Der Autor des »Ascompiler 64« stellt sich vor:

Ich bin 18 Jahre alt und besuche zur Zeit die 12. Klasse des Gymnasiums Gernsheim.

Meine ersten Erfahrungen mit dem Computer machte ich im Informatikunterricht der Mittelstufe. Bald darauf kaufte ich mir einen VC 20, um später dann auf den C 64 umzusteigen.

Der Ascompiler 64 entstand aus der Idee heraus, Routinen, die in Assembler zu kompliziert und in Basic zu langsam sind, in einfachem Basic schreiben zu können, die Programme anschließend zu compilieren und die nun sehr viel schnelleren Programme in andere Basic-Programme einzubauen. Daher habe ich großen Wert darauf gelegt, daß der vom Compiler erzeugte Code extrem kurz und schnell ist, so daß der Ascompiler 64 ein ideales Hilfsmittel für den Basic-Programmierer darstellt, der sich nicht unbedingt mit Maschinensprache herumschlagen will. (Stephan Bartosch)



Dateien im Griff

Mit dieser Basic-Erweiterung wird die Programmierung von guten Dateiverwaltungen enorm erleichtert. Umfangreiche Befehle zur Datenein- und ausgabe, zur Stringverarbeitung und Floppyunterstützung erleichtern die Arbeit ebenso wie problemloses Hoch- und Runterscrollen eines Listings.



Es gibt inzwischen ein fast unüberschaubares Angebot von Basic-Erweiterungen. Diese haben meist den Nachteil, daß von den vielen eingebauten Befehlen nur die Hälfte im »Alltagsgebrauch« benötigt wird. Der Rest geht auf Kosten des Speicherplatzes.

Datawork-Basic wurde auf 22 sehr leistungsfähige Befehle beschränkt. Weiterhin verbraucht es keinerlei Basic-Speicher, da es im Bereich ab $C000 liegt. Besonderer Wert wurde auf einen neuen INPUT-Befehl gelegt: Der normale INPUT-Befehl hat gerade im Zusammenhang mit Dateiverwaltungen einige gravierende Nachteile. Die Cursor-Steuerung wird zum Beispiel nicht abgeschaltet. Dies kann zur Folge haben, daß bei Fehleingaben die gesamte Bildschirmmaske zerstört wird.

Auch läßt sich im Original-Basic des C 64 nicht festlegen, welche Zeichen in der aktuellen Eingabe erlaubt sind. Möchte man zum Beispiel ein Feld zur Namens- und Adressenspeicherung definieren, so sollten zur Eingabe auch nur Groß- und Kleinbuchstaben verwendet werden dürfen. Meistens wird dieses Problem mit sehr umfangreichen und umständlichen GET-Routinen gelöst. Der neue Befehl »FETCH« beseitigt alle genannten Schwierigkeiten und ist obendrein noch wesentlich schneller als eine GET-Abfrage.

Als weiterer, echter Leckerbissen wurde in die Erweiterung eine Centronics-Schnittstelle implementiert. Drucker, die über eine solche Schnittstelle angesprochen werden (zum Beispiel die Epson-Drucker), lassen sich ohne den Kauf von teuren Interfaces ansteuern. Natürlich lassen sich auch weiterhin seriell betriebene Drucker (zum Beispiel die Commodore MPS-Reihe) verwenden. Umgeschaltet wird einfach durch die beiden neuen Befehle »CENTRONICS« und »SERIELL«.

Besonders beachtenswert ist auch der eingebaute »SWAP«-Befehl, der die gefürchtete, zeitintensive »Müllabfuhr« im Stringspeicher, die sogenannte »Garbage-Collection« verhindert, sowie die sehr schnelle Suchroutine, die ein dimensioniertes Feld nach beliebigen Zeichenketten durchsucht.

Eine OLD-Anweisung zur Rettung versehentlich gelöschter Programme, Befehle zur Bildschirmverwaltung und einige Floppy-Kommandos (zum Beispiel Anzeigen des Inhaltsverzeichnisses ohne Programmverlust) runden Datawork-Basic ab.

(Klaus Strunk/tr)

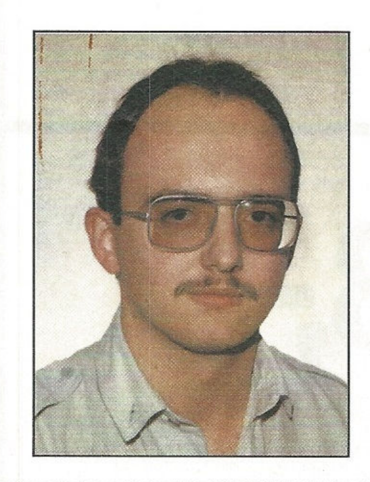
Lebenslauf

Ich bin am 1.6.1962 in Gelsenkirchen geboren und wohne seit 1972 in Bad Waldliesborn bei Lippstadt. Seit zwei Jahren studiere ich Elektrotechnik in Aachen.

Meine Interessen betreffen fast alles, was mit Elektronik zu tun hat. Daneben höre ich sehr gerne Musik und lese alles mögliche.

Nach meinem Abitur 1982 wurde ich zur Bundeswehr eingezogen. Dort bin ich durch einen Freund auf den VC 20 gestoßen und war sofort von den »immensen Möglichkeiten« eines solchen Computers begeistert.

Kurze Zeit später kaufte ich mir selbst einen VC 20 und fing an, munter drauflos zu programmieren. Besonders hatten es mir die Dateiverwaltungen angetan. Nachdem mir Basic zu langsam wurde, fing ich an, kleinere Programme in Maschinensprache zu schreiben.



Als Ende 1983 der C 64 für einen günstigen Preis zu bekommen war, bin ich auf einen C 64 mit Diskettenlaufwerk umgestiegen. Nachdem ich im Laufe der Zeit auch in das Betriebssystem des Computers eingestiegen bin, entschloß ich mich, dem Basic etwas auf die Sprünge zu helfen. So entstand das Datawork-Basic.

(Klaus Strunk)



Datawork-Basic

Mit unserer Anwendung des Monats brauchen Sie nicht länger ein Spitzenprogrammierer zu sein, um gute Dateiverwaltungen selbst zu schreiben. 22 Befehle helfen Ihnen dabei.

Diese Basic-Erweiterung (siehe Listing) wurde in erster Linie entwickelt, um das Programmieren von Dateiprogrammen, besonders derer, die mit Bildschirmmasken arbeiten, zu erleichtern. Um für solche Programme einen möglichst großen Speicherplatz freizulassen, wurde die Erweiterung in den Bereich ab $C000 gelegt, sie belegt somit keinen Basic-Speicher (siehe auch Tabelle 1).

Das Programm arbeitet mit Hypra-Load zusammen. Es werden in der Zeropage einige Speicherstellen verwendet, die normalerweise für den Kassettenbetrieb vorgesehen sind (siehe auch Tabelle 2). Benutzer einer Datasette sollten deshalb ausprobieren, ob ein einwandfreier Betrieb möglich ist.

Das Betriebssystem wird ins RAM kopiert. Deshalb steht für Grafik oder andere Anwendungen das RAM unter dem ROM nicht zur Verfügung.

Start des Programms:

1. Mit Hypra-Load (beziehungsweise allgemein für den Fall, daß das Betriebssystem bereits ins RAM kopiert ist): Zuerst Hypra-Load laden und mit »RUN« starten, dann die Basic-Erweiterung mit »LOAD "DATAWORK 1.1",8,1« laden, und mit »SYS12*4096+6:NEW« ($C006) starten.
2. Ohne Hypra-Load: Die Basic-Erweiterung wie oben laden und mit »SYS12*4096+3:NEW« ($C003) starten.
3. Nach einem Reset: Mit »SYS12*4096« ($C000) starten. Ein Basic-Programm kann danach mit dem OLD-Befehl zurückgeholt werden.

I. Eingabebefehle

FETCH

Format: FETCH "Vorgabestring", "zulässige Zeichen", Länge, Eingabevariable

Oder: FETCH "Numerische Vorgabe", "zulässige Zeichen", Länge, Eingabevariable

»FETCH« ist ein neuer INPUT-Befehl, der diesen vollständig ersetzen kann. Mit »FETCH« wird ein Eingabefeld bestimmter Länge erzeugt, in dem man sich mit den Cursor-Tasten frei bewegen kann. Die Tasten INST und DEL haben ihre normale Bedeutung. Mit der CLR-Taste wird das Eingabefeld gelöscht. Der String oder der numerische Wert »Vorgabe« wird in das Feld geschrieben und kann dann bearbeitet werden. Für »Vorgabe« sind direkte Angaben und Variable jeder Art erlaubt. Der String »zulässige Zeichen« enthält alle Zeichen, die bei der Eingabe erlaubt sein sollen, andere Zeichen werden dann nicht angenommen. Für ganze Zeichengruppen stehen als Abkürzung zur Verfügung:

CRTL-K : Alle Kleinbuchstaben
CRTL-G : Alle Großbuchstaben
CRTL-N : Ziffern von 0 bis 9
CRTL-Z : Alle Zeichen, außer Grafikzeichen und »?« beziehungsweise »*« wegen deren besonderer Bedeutung.

Der Wert »Länge« gibt die Länge des Eingabefeldes an. In der Eingabevariablen wird die Eingabe abgelegt. Es sind sowohl Stringvariable, als auch numerische Variable zulässig. Bei numerischen Variablen wird ein Zahlenwert gemäß dem Basic-Befehl VAL(X$) übergeben. Für den Fall, daß die Eingabe nur Buchstaben enthält, wird also der Wert 0 übergeben. Beispiel: FETCH "Mueller", "{CTRL-K,CTRL-G,CTRL-Z},20,AB$ löscht ein 20 Zeichen langes Eingabefeld und schreibt den Text Mueller hinein. Dann erscheint der Cursor im Feld und der Text kann geändert oder ergänzt werden. Nach RETURN wird die Eingabe an die Variable AB$ übergeben.

— FETCH " ", "12345",2,A — erzeugt ein leeres, zwei Zeichen langes Eingabefeld, in dem nur die Tasten 1 bis 5 gedrückt werden können.

TAKE

Format: TAKE kz, "zulässige Zeichen", Eingabevariable

Der TAKE-Befehl wartet auf eine gedrückte Taste und übergibt sie an die Eingabevariable, ähnlich wie eine GET-Schleife. Bei »TAKE 0,...« wird kein Cursor sichtbar, bei »TAKE 1,...« blinkt der Cursor und die gedrückte Taste wird auf den Bildschirm geschrieben.

Für »zulässige Zeichen« und »Eingabevariable« gilt Gleiches wie bei »FETCH«.

INPUT *

Format: INPUT * Filenummer, Eingabevariable,...

Die Syntax dieses Befehls ist ähnlich wie bei »INPUT #«. »INPUT * « liest jedoch alle Zeichen bis zum nächsten »CHR$(13)« aus einem File. Es können also führende Leerzeichen, Kommata und Doppelpunkte eingelesen werden. »INPUT * « kann maximal 256 Zeichen lesen. Für »Eingabevariable« gilt das gleiche wie bei »FETCH«. Es können beliebig viele Variable angehängt werden.

SETIN

Format: SETIN lfn, Länge, Variable,...

»SETIN« dient dazu, einen Datensatz aus einer Datei zu lesen und ihn gleichzeitig in seine Datenfelder zu zerlegen. »lfn« gibt die logische Filenummer an. Es werden so viele Zeichen aus der Datei gelesen, wie »Länge« angibt. Diese werden dann in die Variable übertragen. Leerzeichen am Ende der einzelnen Datenfelder werden zwar eingelesen, jedoch nicht an die Variablen übergeben. Es können beliebig viele Parameter, bestehend aus Länge und Variable, folgen.

Analog zu »SETIN« gibt es den Befehl »SETOUT« zur Ausgabe eines Datensatzes (siehe weiter unten).

II. Ausgabebefehle

PRINTAT

Format: PRINTAT sp,ze, Ausgabeliste

»sp« und »ze« geben Zeile und Spalte an, in die der Cursor vor der Ausgabe positioniert wird. Die »Ausgabeliste« wird genau wie beim normalen PRINT-Befehl angehängt.

FIELD

Format: FIELD sp,ze, Länge, Zeichen, Farbe, Ausgabeliste

Der FIELD-Befehl positioniert den Cursor an die Stelle »sp« und »ze«, dann wird ein Ausgabefeld der angegebenen Länge mit dem angegebenen Zeichen in angegebener Farbe gefüllt. Danach wird in dieses Feld die Ausgabeliste wie beim PRINT-Befehl ausgegeben. Für »Zeichen« wird der Bildschirmcode (0 bis 255) angegeben. Für »Farbe« sind die Werte 0 bis 15 gültig.

Zum Beispiel: FIELD 10,15,20,32,5,"Text" — löscht ein Ausgabefeld ab Spalte 10, Zeile 15 der Länge 20 und schreibt in grüner Farbe »Text« in das Feld. — FIELD 0,0,40,160,5,"{RVS ON} Überschrift" — läßt in der obersten Zeile einen inversen Balken mit dem Text »Überschrift« erscheinen. Dieser Befehl läßt sich besonders gut für Bildschirmmasken verwenden.

SETOUT
Format: SETOUT lfn, Länge, String,...

»SETOUT« dient dazu, Datensätze in eine Datei zu übertragen, zum Beispiel bei relativen Dateien. »lfn« gibt die logische Filenummer der Datei an. »Länge« gibt die Anzahl der Zeichen an, die übertragen werden. »String« ist eine auszugebende Stringvariable. Ist der String kürzer als »Länge«, so wird er mit Leerzeichen aufgefüllt. Im umgekehrten Fall wird ein »STRING TOO LONG ERROR« ausgegeben. Es können beliebig viele Parameter, bestehend aus Länge und String folgen.

Durch diesen Befehl spart man sich das umständliche Formatieren eines Datensatzes aus seinen einzelnen Feldern.

III. Diskettenbefehle

DIR
Format: DIR KZ, "$", Zeilen
Mit dem DIR-Befehl wird ein Inhaltsverzeichnis der Diskette auf dem Bildschirm ausgegeben.

Wird für »KZ« der Wert 0 angegeben, wird ein einspaltiges Inhaltsverzeichnis ausgegeben, mit dem Wert 1 ein zweispaltiges. Für den String sind Angaben wie beim Laden des Inhaltsverzeichnisses mit dem LOAD-Befehl möglich.
Zum Beispiel: "$" , "$:* =SEQ" und so weiter

»Zeilen« gibt an, wieviele Zeilen zunächst auf dem Bildschirm ausgegeben werden sollen. Danach stoppt die Ausgabe und kann durch Drücken einer beliebigen Taste zeilenweise fortgesetzt werden. Mit der STOP-Taste kann die Ausgabe vorzeitig abgebrochen werden.

CAT
Format: CAT
»CAT« gibt wie »DIR 0,...« ein Inhaltsverzeichnis der Diskette aus, es sind jedoch keine Parameter erforderlich.

DISK
Format: DISK "String"
Mit dem Disk-Befehl wird ein Befehlsstring an den Befehlskanal des Diskettenlaufwerks übergeben.

ERROR
Format: ERROR (numerische Variable, Stringvariable)
Mit dem ERROR-Befehl wird der Fehlerkanal des Diskettenlaufwerks ausgelesen. Die Fehlernummer wird in die numerische Variable, der komplette Fehlerstring in die Stringvariable übergeben.

ERR
Format: ERR
»ERR« gibt den Fehlerkanal des Diskettenlaufwerks direkt auf den Bildschirm aus.

IV. Stringbefehle

SWAP
Format: SWAP String1, String2,...
Mit dem SWAP-Befehl werden die Inhalte der beiden Strings ausgetauscht. Es können weitere Parameter, bestehend aus jeweils zwei zu tauschenden Strings, folgen.

Mit diesem Befehl ist es möglich, die gefürchtete »Garbage Collection« bei umfangreicher Stringverarbeitung, zum Beispiel beim Sortieren, zu vermeiden.

PLACE
Format: PLACE (String1, String2, numerische Variable)
Der PLACE-Befehl stellt fest, ob »String1« irgendwie in »String2« enthalten ist. Ist er nicht enthalten, so wird an die »numerische Variable« der Wert 0, sonst die Position, ab der »String1« in »String2« enthalten ist, übergeben.

STRCOMP
Format: STRCOMP »Wert«
Mit dem STRCOMP-Befehl wird der normale Stringvergleich in Programmen beeinflußt.

Für »Wert« können eingesetzt werden:

1 :	Großbuchstaben werden Kleinbuchstaben gleichgesetzt. Zum Beispiel: "a"="A" ist dann wahr.
64 :	Das Vergleichen von Strings mit einem Stern als Abkürzung ist dann möglich. Zum Beispiel: "dis*" = "diskette" ist dann wahr.
128 :	Das Vergleichen mit einem Fragezeichen als Joker ist möglich. Zum Beispiel: "m??er" = "meier" ist wahr. "m??er" = "mayer" ist auch wahr.

Die Werte können auch addiert werden, um mehrere Funktionen zu nutzen. Zum Beispiel: »STRCOMP 128+64+1« schaltet alle genannten Funktionen ein. Mit »STRCOMP 0« wird der normale Stringvergleich wiederhergestellt.

SEARCH
Format: SEARCH (A$,B$(0,0),P,Q,X%,Y%)
Der SEARCH-Befehl aktiviert die Intellisearch-Routine, die in der 64'er-Ausgabe 3/85 genau beschrieben wird. Man sollte dort auf jeden Fall nachlesen, da die Möglichkeiten der Routine sehr umfangreich sind. Kurze Erklärung:

A$:	Der zu suchende String
B$(0,0):	Das zu durchsuchende Array. Laufvariable ist immer nur die 1. Dimension des Arrays. Startwert nach Wahl, zum Beispiel auch B$(20,2).
P:	Anzahl der zu durchsuchenden Strings
Q:	Startposition der Suche im String
X%:	Rückgabe der Nummer des Strings, wenn gefunden, sonst —1.
Y%:	Rückgabe der Position im String. Wenn nicht gefunden: unverändert.

Zusammenarbeit mit »STRCOMP«: Die Intellisearchroutine hat einen eigenen Programmteil zum Stringvergleich, deshalb ist das Suchen mit »?« und »*« unabhängig von der Einstellung durch »STRCOMP« möglich. Das Gleichsetzen oder Unterscheiden von Groß- und Kleinbuchstaben kann jedoch durch den STRCOMP-Befehl eingestellt werden.

V. Druckersteuerung

CENTRONICS
Format: CENTRONICS (numerische Variable)
Der CENTRONICS-Befehl aktiviert eine Software-Schnittstelle, die über den User-Port mit der Geräteadresse 4 direkt einen Centronics-Drucker anspricht. Es erfolgt eine Commodore-ASCII-Codewandlung, die für den normalen Textbetrieb notwendig ist. Sollte diese Umwandlung einmal unerwünscht sein, so kann sie durch »POKE12*4096+9,255« ($C009) abgeschaltet und durch »POKE12*4096+9,0« wieder eingeschaltet werden. Bei der Initialisierung wird getestet, ob der Drucker ansprechbar, das heißt eingeschaltet und »Online« ist. Ist dies der Fall, so wird in die numerische Variable der Wert 0, andernfalls der Wert 255 übergeben.

SERIELL
Format: SERIELL
Der SERIELL-Befehl leitet die Druckausgabe wieder über den seriellen Bus.

VI. Verschiedene Befehle

FILL
Format: FILL sp1,ze1,sp2,ze2, Code, Farbe
Der FILL-Befehl füllt einen Bildschirmbereich mit einem Zeichen bestimmter Farbe. »sp1« und »ze1« geben Spalte und Zeile der linken oberen Ecke des Bereichs an. »sp2« und »ze2« geben Spalte und Zeile der rechten unteren Ecke an. Für »Code« wird der Bildschirmcode (0 bis 255) des gewählten Zeichens angegeben. Für »Farbe« gelten die Werte 0 bis 15.

TRAP
Format: TRAP KZ
Mit »TRAP 0« wird die Stoptaste abgeschaltet, mit »TRAP 1«

wird sie eingeschaltet. Die Stoptaste kann, wenn sie abgeschaltet ist, mit »GET« oder »TAKE« abgefragt werden.

OLD

Format: OLD

Der OLD-Befehl rettet ein Basic-Programm nach einem »NEW« oder einem Reset, solange der Basic-Speicher nicht weiter verändert worden ist.

SCROLL

Format: SCROLL erste Zeile, letzte Zeile

Der SCROLL-Befehl dient dazu, den Zeilenbereich für das normale Bildschirmscrollen einzuschränken. Er ist jedoch nur mit äußerster Vorsicht zu verwenden, da er leicht zu einem Absturz des Computers führen kann. Der Befehl dient eigentlich nur dazu, zum Beispiel bei der Ausgabe eines Directorys, auf dem Bildschirm eine Überschrift am oberen Bildrand oder einen Text am unteren Rand zu erhalten. Er sollte sofort nach der Ausgabe mit »SCROLL 0,24« rückgängig gemacht werden.

Es ist zu beachten, daß sich der Cursor auf jeden Fall innerhalb des Zeilenbereichs befindet, und daß die ausgegebenen Zeilen nicht länger als 39 Zeichen sind. Dann kann eigentlich nichts schiefgehen.

$C000	Starteinsprung nach Reset
$C003	Starteinsprung mit Verschieben des ROM ins RAM
$C006	Starteinsprung ohne Verschieben
$C009	Flag für Codewandlung der Centronics-schnittstelle
$C00A	Flag für den STRCOMP-Befehl
$C00B bis C024	Tabelle
$C024 bis C052	Adressen der Basicbefehle (—1)
$C066 bis C06C	Zeiger
$C06C bis CE8A	Programm
$CEBA bis CF00	Tabelle der Befehlswörter
$CF00 bis CFFF	Puffer
$CF00 bis CF5B	Startprogramm
$CF5B bis CFB8	Tabellen

Zeiger im Programm:

$C1E5	LDA #C9
$C2C4	LDA #CA
$C3D9	LDA #CB
$CC07	LDA #C0

Zeropage-Adressen:
144, 146, 150, 155, 156, 168 bis 179, 189 bis 191, 248 bis 254

Tabelle 2. Die wichtigsten Adressen von »Datawork-Basic«

VII. Verschiedenes

Die Geräte-Adresse für den LOAD-Befehl ist nicht mehr auf 1, sondern auf 8 voreingestellt. Sie braucht deshalb beim Laden und Speichern nicht unbedingt angegeben werden.

Das Editieren von Programmen wird durch eine komfortable Listroutine erleichtert. Das Listing kann mit den Cursortasten über den Bildschirm hinauf- und hinuntergescrollt werden.

Die Funktion ASC("") liefert den Wert 0 und nicht wie bisher einen »ILLEGAL QUANTITY ERROR«.

Der Tastatur-Repeat ist für alle Tasten voreingestellt. Er kann durch POKE650,0 rückgängig gemacht werden.

(Klaus Strunk/tr)

VIII. Verschieben des Programms

Das Verschieben des Programms in einen anderen Speicherbereich (hier $9000 bis $9C00) soll am Beispiel des SMON beschrieben werden.

1. Verschieben nach 9000
   W C000 D000 9000
2. Programmteile anpassen
   V C000 D000 9000 9000 9009
   V C000 D000 9000 906C 9E8A
   V C000 D000 9000 9F00 9F5B
3. Anpassen der Zeiger

Zeigertabelle anzeigen: M 9024 9052
Alle Zeiger, die sich auf C... beziehen, in 9... ändern.
Zeigertabelle 9066 bis 906B anzeigen: M 9066 906B
Alle Zeiger, die sich auf C... beziehen, in 9... ändern.
Einzelne Zeiger im Programm:
D 91E5 91E6
D 92C4 92C5
D 93D9 93DA
D 9C07 9C09
Zeiger in 9... ändern

4. Startadressen:

Von den Startadressen (im Vorwort) ist der Wert 3*4096 abzuziehen.

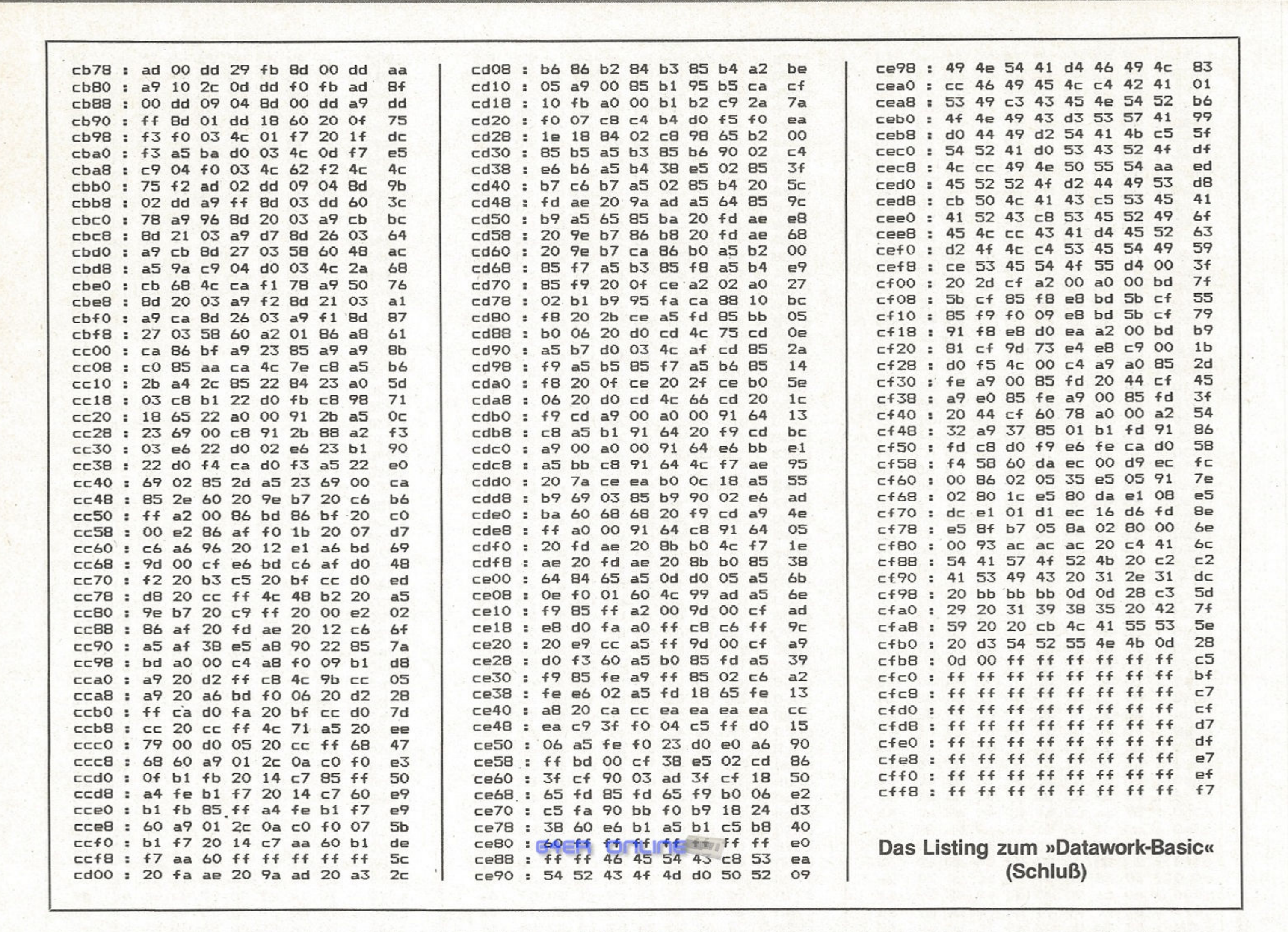
Tabelle 1. So verschieben Sie »Datawork-Basic« in einen anderen Speicherbereich


```
programm : datawork 1.1  c000 d000
-------------------------------------
c000 : 4c 00 c4 4c 00 cf 4c 03  bd
c008 : cf 00 00 21 23 24 25 26  30
c010 : 27 28 29 2b 2c 2d 2e 2f  3e
c018 : 3a 3b 3c 3d 3e 40 5b 5c  b3
c020 : 5d 5e 5f 24 3d c4 21 c7  17
c028 : 5f c7 6c c7 d7 c7 79 c0  a2
c030 : df ca 3d c8 6c c8 6f c9  3b
c038 : 8e c9 bb c9 d0 c9 fb c9  b2
c040 : 42 ca 69 ca ff cc e4 cb  2c
c048 : fb cb 3b ca 0e cc 4a cc  5b
c050 : 7e cc ff ff ff ff ff ff  34
c058 : ff ff ff ff ff ff ff ff  57
c060 : ff ff ff ff ff a0 fc c0  d9
c068 : 80 c0 bb c0 78 a2 05 bd  7b
c070 : 66 c0 9d 04 03 ca 10 f7  d5
c078 : 58 60 78 20 53 e4 58 60  a1
c080 : 10 32 c9 ff f0 2e 24 0f  4b
c088 : 30 2a c9 cc 30 2a 38 e9  e2
c090 : cb aa 84 49 a0 ff ea ca  46
c098 : f0 0a ea c8 b9 8a ce 10  ac
c0a0 : fa 30 f4 ea c8 b9 8a ce  6f
c0a8 : 30 06 20 47 ab d0 f5 ea  bb
c0b0 : 4c ef a6 ea 4c f3 a6 ea  d0
c0b8 : 4c 24 a7 20 73 00 20 c4  45
c0c0 : c0 4c ae a7 f0 0c e9 80  5f
c0c8 : 90 0c c9 23 b0 0c 4c f7  c2
c0d0 : a7 ea 4c 2b a8 ea 4c a5  c3
c0d8 : a9 ea c9 4b 30 06 d0 08  59
c0e0 : 4c 12 a8 ea 4c 08 af ea  56
c0e8 : c9 7f f0 f8 38 e9 4c 0a  e4
c0f0 : a8 b9 25 c0 48 b9 24 c0  3b
c0f8 : 48 4c 73 00 a6 7a a0 04  0c
c100 : 84 0f bd 00 02 10 07 c9  cc
c108 : ff f0 41 e8 d0 f4 c9 20  09
c110 : f0 3a 85 08 c9 22 f0 59  a4
c118 : 24 0f 70 30 c9 3f d0 04  c8
c120 : a9 99 d0 28 c9 30 90 04  37
c128 : c9 3c 90 20 4c 96 c1 84  c1
c130 : 71 a0 00 84 0b 88 86 7a  86
c138 : ca c8 e8 bd 00 02 38 f9  3d
c140 : 9e a0 f0 f5 c9 80 d0 30  6d
c148 : 05 0b a4 71 e8 c8 99 fb  5d
c150 : 01 b9 fb 01 f0 36 38 e9  c2
c158 : 3a f0 04 c9 49 d0 02 85  73
c160 : 0f 38 e9 55 d0 9c 85 08  c8
c168 : bd 00 02 f0 df c5 08 f0  f2
c170 : db c8 99 fb 01 e8 d0 f0  12
c178 : a6 7a e6 0b c8 b9 9d a0  88
c180 : 10 fa b9 9e a0 d0 b4 bd  2e
c188 : 00 02 10 be 99 fd 01 c6  80
c190 : 7b a9 ff 85 7a 60 84 71  30
c198 : a0 00 84 0b 88 86 7a ca  f7
c1a0 : c8 e8 bd 00 02 38 f9 8a  2b
c1a8 : ce f0 f5 c9 80 d0 08 05  5e
c1b0 : 0b 18 69 4c 4c 4a c1 a6  17
c1b8 : 7a e6 0b c8 b9 89 ce 10  c4
c1c0 : fa b9 8a ce d0 dc a4 71  7d
c1c8 : a6 7a 4c 2f c1 ff ff ff  c0
c1d0 : ff ff ff ff ff ff ff ff  cf
c1d8 : ff ff ff ff ff ff ff ff  d7
c1e0 : a9 eb 8d 02 03 a9 c1 8d  c2
c1e8 : 03 03 60 a5 d6 85 c9 a5  46
c1f0 : d3 85 ca 20 16 e7 a9 00  84
c1f8 : 8d ef c2 a5 c6 85 cc 8d  c9
c200 : 92 02 f0 f7 78 a5 cf f0  a4
c208 : 0c a5 ce ae 87 02 a0 00  7b
c210 : 84 cf 20 13 ea 20 b4 e5  35
c218 : c9 0d d0 03 4c c4 c2 a6  40
c220 : d4 d0 d0 a6 d8 d0 cc c9  40
c228 : 11 f0 43 c9 91 d0 c4 8d  89
c230 : f0 c2 a5 d6 d0 32 ad ef  fb
c238 : c2 d0 0c 20 22 c3 90 bb  63
c240 : 20 63 c3 f0 b6 d0 0d a5  92
c248 : f7 85 5f a5 f8 85 60 20  0c
c250 : 66 c3 f0 a7 a9 01 8d ef  82
c258 : c2 20 f1 c2 20 66 e5 20  0c
c260 : be c3 20 66 e5 4c fb c1  09
c268 : ad f0 c2 4c f3 c1 8d f0  2d
```

Das Listing zum »Datawork-Basic« geben Sie bitte mit dem MSE ein

```
c270 : c2 a5 d6 c9 18 90 f1 ad  1d
c278 : ef c2 d0 0c 20 32 c3 90  42
c280 : e7 20 9e c3 f0 e2 d0 0d  1b
c288 : a5 f7 85 5f a5 f8 85 60  6f
c290 : 20 a7 c3 f0 d3 a9 01 8d  3c
c298 : ef c2 a2 17 b5 d9 30 0b  75
c2a0 : e8 e0 19 90 f7 20 ea e8  4f
c2a8 : 4c 9a c2 20 40 c3 c9 20  7f
c2b0 : d0 ee a0 00 20 0c e5 20  5a
c2b8 : be c3 a2 18 a0 00 20 0c  a6
c2c0 : e5 4c fb c1 a9 c2 48 a9  27
c2c8 : e1 48 78 a6 c6 f0 09 bd  54
c2d0 : 76 02 9d 77 02 ca d0 f7  47
c2d8 : a9 0d 8d 77 02 e6 c6 20  0d
c2e0 : d6 e5 a8 a9 a4 48 a9 85  46
c2e8 : 48 a2 00 98 4c 65 a5 00  1b
c2f0 : 91 a2 19 ca 20 f0 e9 e0  65
c2f8 : 00 f0 0c bd ef ec 85 ac  01
c300 : b5 d8 20 c8 e9 30 ec 20  57
c308 : ff e9 a2 18 b5 d9 29 7f  75
c310 : b4 d8 10 02 09 80 95 d9  13
c318 : ca 10 f1 a5 d9 09 80 85  0e
c320 : d9 60 a2 00 b5 d9 10 03  42
c328 : 20 51 c3 e8 e0 19 90 f4  02
c330 : 18 60 a2 18 b5 d9 10 03  94
c338 : 20 51 c3 ca 10 f6 18 60  25
c340 : 29 03 0d 88 02 85 7b bd  f5
c348 : f0 ec 85 7a a0 00 b1 7a  25
c350 : 60 20 40 c3 c9 3a b0 0a  8e
c358 : c9 30 90 06 18 20 6b a9  a2
c360 : 68 68 60 20 13 a6 a6 60  da
c368 : a5 5f e4 2c d0 04 c5 2b  16
c370 : f0 2b ca 85 f7 86 f8 a0  32
c378 : ff c8 b1 f7 d0 fb c8 b1  ba
c380 : f7 c5 5f d0 f5 c8 b1 f7  a8
c388 : c5 60 d0 ee 18 88 98 65  82
c390 : f7 85 5f 85 f7 a5 f8 69  36
c398 : 00 85 60 85 f8 60 20 13  5d
c3a0 : a6 a6 60 a5 5f 90 0a a0  4a
c3a8 : 01 b1 5f f0 10 18 20 38  2a
c3b0 : a6 85 5f 85 f7 86 60 86  e4
c3b8 : f8 a0 01 b1 5f 60 a0 01  f4
c3c0 : 8c 92 02 84 0f b1 5f 48  33
c3c8 : a9 00 91 5f a5 5f 85 fb  25
c3d0 : a5 60 85 fc a9 e8 8d 00  be
c3d8 : 03 a9 c3 8d 01 03 c8 b1  01
c3e0 : 5f aa c8 b1 5f 4c e8 a6  46
c3e8 : a0 01 68 91 fb a9 8b 8d  ab
c3f0 : 00 03 a9 e3 8d 01 03 a9  99
c3f8 : 00 85 d4 85 c7 85 d8 60  6d
c400 : a9 35 85 01 20 6c c0 20  6e
c408 : e0 c1 20 a0 e5 a9 08 20  f1
c410 : d2 ff a9 94 8d 00 a0 a9  8d
c418 : 05 8d 86 02 a9 73 a0 e4  48
c420 : 20 1e ab a5 37 38 e5 2b  12
c428 : aa a5 38 e5 2c 20 cd bd  e6
c430 : a9 60 a0 e4 20 1e ab a9  c3
c438 : 00 85 0d 4c f7 b8 38 20  2e
c440 : f0 ff 86 b1 84 b0 20 07  64
c448 : c6 20 9e ad 24 0d 30 13  0d
c450 : 20 dd bd a2 00 bd 00 01  12
c458 : c9 00 f0 17 9d 00 cf e8  2b
c460 : 4c 55 c4 20 15 c6 a4 a8  f7
c468 : 88 30 08 b1 a9 99 00 cf  48
c470 : 4c 68 c4 20 fd ae 20 12  1f
c478 : c6 20 fd ae 20 9e b7 86  86
c480 : bd a9 00 85 bf a9 01 85  1b
c488 : 96 a9 00 85 92 20 65 c6  f1
c490 : a5 96 f0 0c 20 34 c6 a5  48
c498 : 92 c5 bd f0 03 20 58 c6  ba
c4a0 : a5 c6 f0 fc 20 e4 ff 85  b8
c4a8 : 9b a6 bf f0 06 20 30 c5  52
c4b0 : 4c 8d c4 c9 93 d0 06 20  45
c4b8 : 07 c6 4c 89 c4 c9 14 d0  f3
c4c0 : 1e a4 92 f0 17 a9 20 a6  80
c4c8 : bd 9d 00 cf 88 c8 b9 00  04
c4d0 : cf 88 99 00 cf c8 c4 bd  1b
c4d8 : 30 f3 c6 92 4c 8d c4 c9  de
c4e0 : 9d d0 09 a5 92 f0 02 c6  23
c4e8 : 92 4c 8d c4 c9 0d d0 03  ea
c4f0 : 4c a0 c5 a6 bd e4 92 d0  c1
c4f8 : 03 4c 8d c4 c9 94 d0 1f  e0
c500 : a4 bd 88 a9 20 d9 00 cf  4b
c508 : d0 12 88 b9 00 cf c8 99  0f
c510 : 00 cf 88 c4 92 d0 f3 a9  85
c518 : 20 99 00 cf 4c 8d c4 a9  96
c520 : 1d c5 9b d0 05 e6 92 4c  8b
c528 : 8d c4 20 30 c5 4c 8d c4  a4
c530 : a9 0b 20 78 c6 90 0c a5  e2
c538 : 9b 30 08 20 13 b1 90 03  f8
c540 : 4c 8c c5 a9 07 20 78 c6  5a
c548 : 90 0e a5 9b 10 0a 29 7f  b1
c550 : 20 13 b1 90 03 4c 8c c5  c9
c558 : a9 0e 20 78 c6 90 10 a5  9c
c560 : 9b c9 30 90 0a e9 3a 38  47
c568 : e9 c6 90 03 4c 8c c5 a9  cc
c570 : 1a 20 78 c6 90 0e a5 9b  d8
c578 : a2 00 dd 0b c0 f0 0d e8  8d
c580 : e0 18 d0 f6 a5 9b 20 78  28
c588 : c6 b0 01 60 a4 92 c4 bd  60
c590 : f0 0b a5 9b 99 00 cf e6  89
c598 : 92 a5 bf d0 01 60 68 68  8c
c5a0 : 20 65 c6 a5 96 f0 0c 20  3a
c5a8 : 34 c6 a5 bf d0 05 a9 0d  97
c5b0 : 20 d2 ff 20 fd ae 20 8b  2a
c5b8 : b0 85 49 84 4a a5 bf d0  80
c5c0 : 10 a6 bd f0 0c ca bd 00  bf
c5c8 : cf c9 20 d0 04 c6 bd b0  6d
c5d0 : f0 24 0d 10 20 a5 bd 85  49
c5d8 : 61 f0 0f 20 7d b4 a0 00  79
c5e0 : b9 00 cf 91 62 c8 c4 bd  ba
c5e8 : d0 f6 a0 02 b9 61 00 91  65
c5f0 : 49 88 10 f8 60 a9 00 85  ff
c5f8 : 22 a9 cf 85 23 a5 bd 20  2a
c600 : b0 b7 a5 0e 4c c2 a9 a2  7e
c608 : 00 a9 20 9d 00 cf e8 d0  5c
c610 : fa 60 20 9e ad 20 a3 b6  ee
c618 : a0 02 b1 64 99 a8 00 88  a2
c620 : 10 f8 60 20 9e ad 20 a3  e8
c628 : b6 a0 02 b1 64 99 fc 00  ec
c630 : 88 10 f8 60 a6 b1 a4 b0  f6
c638 : 18 20 f0 ff a2 00 e4 92  7f
c640 : d0 0a 38 20 f0 ff 86 b3  b8
c648 : 84 b2 a6 92 bd 00 cf 20  7d
c650 : d2 ff e8 e4 bd d0 e7 60  bb
c658 : a6 b3 a4 b2 18 20 f0 ff  9d
c660 : a9 00 85 cc 60 a5 cc d0  0c
c668 : 0e a5 ce ae 87 02 a0 00  dd
c670 : 84 cf 20 13 ea e6 cc 60  20
c678 : a4 a8 f0 0c a0 00 d1 a9  d3
c680 : f0 06 c8 c4 a8 d0 f7 18  5f
c688 : 60 85 bd 2c 0a c0 50 27  d6
c690 : a8 f0 0f 88 b1 6c c9 2a  7f
c698 : d0 08 84 bd c4 61 b0 02  63
c6a0 : 84 61 a4 61 f0 0f 88 b1  37
c6a8 : 62 c9 2a d0 08 84 61 c4  47
c6b0 : bd b0 02 84 bd a5 bd aa  2c
c6b8 : 38 e5 61 f0 08 a9 01 90  4c
c6c0 : 04 a6 61 a9 ff 85 66 a0  ac
c6c8 : ff e8 c8 ca d0 07 a6 66  73
c6d0 : 30 23 18 90 20 b1 62 20  03
c6d8 : f8 c6 b0 ee 20 05 c7 85  92
c6e0 : 9c b1 6c 20 f8 c6 b0 e2  c2
c6e8 : 20 05 c7 c5 9c f0 db a2  3b
c6f0 : ff b0 02 a2 01 4c 72 b0  ba
c6f8 : c9 3f d0 07 2c 0a c0 10  ac
c700 : 02 38 60 18 60 c9 80 90  b1
c708 : 18 48 a9 01 2c 0a c0 d0  87
c710 : 02 68 60 68 c9 c1 90 09  6a
c718 : e9 db 38 e9 25 90 02 29  6b
c720 : 7f 60 20 9e b7 a0 02 8e  51
c728 : 0a c0 8a d0 0a b9 44 c7  5e
c730 : 99 48 b0 88 10 f7 60 b9  e0
c738 : 41 c7 99 48 b0 88 10 f7  4c
c740 : 60 4c 89 c6 aa 38 e5 20  46
c748 : 9e b7 86 b2 e0 28 b0 0d  e6
c750 : 20 fd ae 20 9e b7 e0 19  7c
c758 : b0 03 a4 b2 60 4c 48 b2  f8
c760 : 20 47 c7 18 20 f0 ff 20  e2
c768 : fd ae 4c a0 aa 38 20 f0  b2
c770 : ff 86 b1 84 b0 20 47 c7  68
c778 : 86 ac 84 ab 20 fd ae 20  d8
c780 : 47 c7 86 ae 84 ad 20 fd  54
c788 : ae 20 9e b7 86 9c 20 fd  af
c790 : ae 20 9e b7 86 bf a6 ac  47
c798 : a4 ab 18 20 f0 ff 20 24  f4
c7a0 : ea a5 9c a6 bf 20 1c ea  9c
c7a8 : a4 d3 c0 27 f0 08 c4 ad  09
c7b0 : f0 04 e6 d3 d0 eb a6 ac  37
c7b8 : e0 18 f0 14 e4 ae f0 10  0b
c7c0 : e6 ac a4 ab a6 ac 18 20  0b
c7c8 : f0 ff 20 24 ea 4c a1 c7  6c
c7d0 : a6 b1 a4 b0 18 4c f0 ff  36
c7d8 : 20 47 c7 86 ac 84 ab 20  3c
c7e0 : fd ae 20 9e b7 86 af 20  bf
c7e8 : fd ae 20 9e b7 86 9c 20  7b
c7f0 : fd ae 20 9e b7 86 bf 20  0f
c7f8 : fd ae a6 ac a4 ab 18 20  d4
c800 : f0 ff 20 24 ea 98 18 65  1b
c808 : af b0 10 85 96 38 e9 28  e7
c810 : 90 05 f0 03 e8 d0 f6 e0  72
c818 : 19 90 03 4c 48 b2 a5 9c  ae
c820 : a6 bf 20 1c ea e6 d3 a5  b2
c828 : d3 c5 96 d0 f1 a5 bf 8d  04
c830 : 86 02 a6 ac a4 ab 18 20  3f
c838 : f0 ff c8 4c a0 aa 20 9e  01
c840 : ad 20 8f ad a5 64 85 f8  1c
c848 : a5 65 85 f9 20 fd ae 20  2d
c850 : 9e ad 20 8f ad a0 02 b1  0a
c858 : f8 aa b1 64 91 f8 8a 91  cc
c860 : 64 88 10 f3 20 06 e2 20  89
c868 : fd ae 4c 3e c8 20 9e b7  0f
c870 : 86 bf 20 fd ae 20 12 c6  5f
c878 : 20 fd ae 20 9e b7 86 bd  84
c880 : a9 00 85 90 85 b0 85 b1  f4
c888 : 85 96 a5 a8 a6 a9 a4 aa  76
c890 : 20 bd ff a9 08 85 ba a9  af
c898 : 60 85 b9 20 d5 f3 a5 ba  36
c8a0 : 20 b4 ff a5 b9 20 96 ff  c6
c8a8 : a5 90 d0 63 a0 03 84 b7  d9
c8b0 : 20 a5 ff 85 c3 20 a5 ff  27
c8b8 : 85 c4 a4 90 d0 51 a4 b7  74
c8c0 : 88 d0 eb a5 b1 f0 10 a5  8e
c8c8 : bf d0 4c a6 c3 a5 c4 20  94
c8d0 : cd bd a9 20 20 d2 ff e6  51
c8d8 : b1 20 a5 ff a6 90 d0 2f  93
c8e0 : c9 00 d0 f0 20 e4 ff c9  b8
c8e8 : 03 f0 1f c6 bd d0 0d a9  ee
c8f0 : 01 85 bd 20 e4 ff f0 fb  31
c8f8 : c9 03 f0 0e a9 0d 20 d2  6a
c900 : ff a9 00 85 b0 a0 02 4c  35
c908 : ae c8 a9 0d 20 d2 ff 20  ff
c910 : 42 f6 a9 00 85 91 60 a9  f1
c918 : 00 85 ab 20 a5 ff a6 90  e0
c920 : d0 ed c9 22 d0 0a e6 96  c4
c928 : a6 96 e0 02 f0 0e d0 eb  2c
c930 : a6 96 f0 e7 20 d2 ff e6  c1
c938 : ab 4c 1b c9 a9 00 85 96  e7
c940 : 20 a5 ff a6 90 d0 c8 c9  4e
c948 : 00 d0 f5 e6 b0 a5 b0 c9  99
c950 : 02 f0 91 a9 13 38 e5 ab  46
c958 : aa a9 20 20 d2 ff ca d0  dd
c960 : fa a9 3a 20 d2 ff a9 20  d5
c968 : 20 d2 ff a0 02 4c ae c8  d4
c970 : 38 20 f0 ff 86 b1 84 b0  5e
c978 : 20 9e b7 86 96 20 07 c6  ba
c980 : 20 fd ae 20 12 c6 a9 01  4e
c988 : 85 bf 85 bd 4c 89 c4 20  6a
c990 : 9e b7 8a f0 06 a9 31 8d  58
c998 : 14 03 60 a9 34 8d 14 03  81
c9a0 : 60 20 9e b7 e0 19 b0 11  6b
c9a8 : 86 ac 20 fd ae 20 9e b7  22
c9b0 : e0 19 b0 05 86 ae a4 ac  b3
c9b8 : 60 4c 48 b2 20 a1 c9 8e  fa
c9c0 : 04 e9 8e 70 e9 e8 8e 82  90
c9c8 : e8 8e c5 e8 88 8c f7 e8  24
c9d0 : 60 20 9e b7 20 c6 ff a2  5c
c9d8 : 00 86 bd e8 86 bf 20 12  b3
c9e0 : e1 c9 0d f0 0f a6 bd 9d  60
c9e8 : 00 cf e6 bd d0 f0 20 cc  f0
c9f0 : ff 4c 71 a5 20 b3 c5 20  1d
c9f8 : bf cc d0 db a9 00 85 bd  f9
ca00 : 85 96 20 fa ae a9 08 85  9b
ca08 : ba 20 b4 ff a9 6f 85 b9  9f
ca10 : 20 96 ff 20 a5 ff 24 96  97
ca18 : 30 1a c9 0d f0 0a a6 bd  df
ca20 : 9d 00 cf e6 bd 4c 13 ca  ae
ca28 : 20 ab ff 20 b6 c5 20 b3  a3
ca30 : c5 4c f7 ae 20 d2 ff c9  1b
ca38 : 0d d0 d8 60 a9 ff 85 96  cd
ca40 : 4c 05 ca 20 12 c6 a5 a8  05
ca48 : c9 00 f0 1d a9 08 85 ba  57
ca50 : 20 b1 ff a9 6f 85 b9 20  c8
ca58 : 93 ff a0 00 b1 a9 20 a8  4d
ca60 : ff c8 c4 a8 d0 f6 20 ae  ac
ca68 : ff 60 20 fa ae 20 12 c6  c0
ca70 : 20 fd ae 20 23 c6 20 fd  23
ca78 : ae 20 8b b0 85 49 84 4a  78
ca80 : 20 f7 ae 24 0d 10 05 a2  77
ca88 : 16 4c 37 a4 a5 fc 38 e5  16
ca90 : a8 90 2f c5 fc f0 2b 85  14
ca98 : 96 aa a9 00 85 92 a0 00  5d
caa0 : e6 92 b1 a9 d1 fd d0 0f  df
caa8 : c8 c4 a8 d0 f5 a4 92 20  25
cab0 : a2 b3 a5 0e 4c c2 a9 e6  a6
cab8 : fd d0 02 e6 fe a0 00 ca  05
cac0 : 10 dc 20 f7 b8 a5 0e 4c  cf
cac8 : c2 a9 ad 0d dd 20 b2 cb  ad
cad0 : 85 96 a9 04 0d 00 dd 8d  ef
cad8 : 00 dd a9 ff 20 06 cb 60  53
cae0 : 20 c0 cb 20 ca ca 20 fa  d1
cae8 : ae 20 8b b0 85 49 84 4a  e8
caf0 : 20 f7 ae 24 0d 10 05 a2  e7
caf8 : 16 4c 37 a4 a4 96 20 a2  5b
cb00 : b3 a5 0e 4c c2 a9 8d 01  45
cb08 : dd a9 fb 2d 00 dd 8d 00  83
cb10 : dd a2 00 ca d0 fd ad 0d  65
cb18 : dd 29 10 f0 04 a9 00 85  45
cb20 : 96 a9 04 0d 00 dd 8d 00  53
cb28 : dd 60 20 b2 cb a9 04 2d  08
cb30 : 00 dd d0 03 20 80 cb 68  b9
cb38 : 48 2c 09 c0 30 2f c9 41  17
cb40 : 90 2b c9 df b0 27 c9 5f  fe
cb48 : f0 23 c9 60 f0 1f c9 7f  76
cb50 : 30 07 c9 c1 b0 03 4c 6d  de
cb58 : cb c9 5f b0 05 69 20 4c  aa
cb60 : 6d cb c9 7f b0 05 69 60  af
cb68 : 4c 6d cb e9 80 20 72 cb  05
cb70 : 68 60 8d 01 dd ad 0d dd  c7
```

»Datawork-Basic«



```
cb78 : ad 00 dd 29 fb 8d 00 dd   aa
cb80 : a9 10 2c 0d dd f0 fb ad   8f
cb88 : 00 dd 09 04 8d 00 dd a9   dd
cb90 : ff 8d 01 dd 18 60 20 0f   75
cb98 : f3 f0 03 4c 01 f7 20 1f   dc
cba0 : f3 a5 ba d0 03 4c 0d f7   e5
cba8 : c9 04 f0 03 4c 62 f2 4c   4c
cbb0 : 75 f2 ad 02 dd 09 04 8d   9b
cbb8 : 02 dd a9 ff 8d 03 dd 60   3c
cbc0 : 78 a9 96 8d 20 03 a9 cb   bc
cbc8 : 8d 21 03 a9 d7 8d 26 03   64
cbd0 : a9 cb 8d 27 03 58 60 48   ac
cbd8 : a5 9a c9 04 d0 03 4c 2a   68
cbe0 : cb 68 4c ca f1 78 a9 50   76
cbe8 : 8d 20 03 a9 f2 8d 21 03   a1
cbf0 : a9 ca 8d 26 03 a9 f1 8d   87
cbf8 : 27 03 58 60 a2 01 86 a8   61
cc00 : ca 86 bf a9 23 85 a9 a9   8b
cc08 : c0 85 aa ca 4c 7e c8 a5   b6
cc10 : 2b a4 2c 85 22 84 23 a0   5d
cc18 : 03 c8 b1 22 d0 fb c8 98   71
cc20 : 18 65 22 a0 00 91 2b a5   0c
cc28 : 23 69 00 c8 91 2b 88 a2   f3
cc30 : 03 e6 22 d0 02 e6 23 b1   90
cc38 : 22 d0 f4 ca d0 f3 a5 22   e0
cc40 : 69 02 85 2d a5 23 69 00   ca
cc48 : 85 2e 60 20 9e b7 20 c6   b6
cc50 : ff a2 00 86 bd 86 bf 20   c0
cc58 : 00 e2 86 af f0 1b 20 07   d7
cc60 : c6 a6 96 20 12 e1 a6 bd   69
cc68 : 9d 00 cf e6 bd c6 af d0   48
cc70 : f2 20 b3 c5 20 bf cc d0   ed
cc78 : d8 20 cc ff 4c 48 b2 20   a5
cc80 : 9e b7 20 c9 ff 20 00 e2   02
cc88 : 86 af 20 fd ae 20 12 c6   6f
cc90 : a5 af 38 e5 a8 90 22 85   7a
cc98 : bd a0 00 c4 a8 f0 09 b1   d8
cca0 : a9 20 d2 ff c8 4c 9b cc   05
cca8 : a9 20 a6 bd f0 06 20 d2   28
ccb0 : ff ca d0 fa 20 bf cc d0   7d
ccb8 : cc 20 cc ff 4c 71 a5 20   ee
ccc0 : 79 00 d0 05 20 cc ff 68   47
ccc8 : 68 60 a9 01 2c 0a c0 f0   e3
ccd0 : 0f b1 fb 20 14 c7 85 ff   50
ccd8 : a4 fe b1 f7 20 14 c7 60   e9
cce0 : b1 fb 85 ff a4 fe b1 f7   e9
cce8 : 60 a9 01 2c 0a c0 f0 07   5b
ccf0 : b1 f7 20 14 c7 aa 60 b1   de
ccf8 : f7 aa 60 ff ff ff ff ff   5c
cd00 : 20 fa ae 20 9a ad 20 a3   2c
cd08 : b6 86 b2 84 b3 85 b4 a2   be
cd10 : 05 a9 00 85 b1 95 b5 ca   cf
cd18 : 10 fb a0 00 b1 b2 c9 2a   7a
cd20 : f0 07 c8 c4 b4 d0 f5 f0   ea
cd28 : 1e 18 84 02 c8 98 65 b2   00
cd30 : 85 b5 a5 b3 85 b6 90 02   c4
cd38 : e6 b6 a5 b4 38 e5 02 85   3f
cd40 : b7 c6 b7 a5 02 85 b4 20   5c
cd48 : fd ae 20 9a ad a5 64 85   9c
cd50 : b9 a5 65 85 ba 20 fd ae   e8
cd58 : 20 9e b7 86 b8 20 fd ae   68
cd60 : 20 9e b7 ca 86 b0 a5 b2   00
cd68 : 85 f7 a5 b3 85 f8 a5 b4   e9
cd70 : 85 f9 20 0f ce a2 02 a0   27
cd78 : 02 b1 b9 95 fa ca 88 10   bc
cd80 : f8 20 2b ce a5 fd 85 bb   05
cd88 : b0 06 20 d0 cd 4c 75 cd   0e
cd90 : a5 b7 d0 03 4c af cd 85   2a
cd98 : f9 a5 b5 85 f7 a5 b6 85   14
cda0 : f8 20 0f ce 20 2f ce b0   5e
cda8 : 06 20 d0 cd 4c 66 cd 20   1c
cdb0 : f9 cd a9 00 a0 00 91 64   13
cdb8 : c8 a5 b1 91 64 20 f9 cd   bc
cdc0 : a9 00 a0 00 91 64 e6 bb   e1
cdc8 : a5 bb c8 91 64 4c f7 ae   95
cdd0 : 20 7a ce ea b0 0c 18 a5   55
cdd8 : b9 69 03 85 b9 90 02 e6   ad
cde0 : ba 60 68 68 20 f9 cd a9   4e
cde8 : ff a0 00 91 64 c8 91 64   05
cdf0 : 20 fd ae 20 8b b0 4c f7   1e
cdf8 : ae 20 fd ae 20 8b b0 85   38
ce00 : 64 84 65 a5 0d d0 05 a5   6b
ce08 : 0e f0 01 60 4c 99 ad a5   6e
ce10 : f9 85 ff a2 00 9d 00 cf   ad
ce18 : e8 d0 fa a0 ff c8 c6 ff   9c
ce20 : 20 e9 cc a5 ff 9d 00 cf   a9
ce28 : d0 f3 60 a5 b0 85 fd a5   39
ce30 : f9 85 fe a9 ff 85 02 c6   a2
ce38 : fe e6 02 a5 fd 18 65 fe   13
ce40 : a8 20 ca cc ea ea ea ea   cc
ce48 : ea c9 3f f0 04 c5 ff d0   15
ce50 : 06 a5 fe f0 23 d0 e0 a6   90
ce58 : ff bd 00 cf 38 e5 02 cd   86
ce60 : 3f cf 90 03 ad 3f cf 18   50
ce68 : 65 fd 85 fd 65 f9 b0 06   e2
ce70 : c5 fa 90 bb f0 b9 18 24   d3
ce78 : 38 60 e6 b1 a5 b1 c5 b8   40
ce80 : 60 ff ff ff ff ff ff ff   e0
ce88 : ff ff 46 45 54 43 c8 53   ea
ce90 : 54 52 43 4f 4d d0 50 52   09
ce98 : 49 4e 54 41 d4 46 49 4c   83
cea0 : cc 46 49 45 4c c4 42 41   01
cea8 : 53 49 c3 43 45 4e 54 52   b6
ceb0 : 4f 4e 49 43 d3 53 57 41   99
ceb8 : d0 44 49 d2 54 41 4b c5   5f
cec0 : 54 52 41 d0 53 43 52 4f   df
cec8 : 4c cc 49 4e 50 55 54 aa   ed
ced0 : 45 52 52 4f d2 44 49 53   d8
ced8 : cb 50 4c 41 43 c5 53 45   41
cee0 : 41 52 43 c8 53 45 52 49   6f
cee8 : 45 4c cc 43 41 d4 45 52   63
cef0 : d2 4f 4c c4 53 45 54 49   59
cef8 : ce 53 45 54 4f 55 d4 00   3f
cf00 : 20 2d cf a2 00 a0 00 bd   7f
cf08 : 5b cf 85 f8 e8 bd 5b cf   55
cf10 : 85 f9 f0 09 e8 bd 5b cf   79
cf18 : 91 f8 e8 d0 ea a2 00 bd   b9
cf20 : 81 cf 9d 73 e4 e8 c9 00   1b
cf28 : d0 f5 4c 00 c4 a9 a0 85   2d
cf30 : fe a9 00 85 fd 20 44 cf   45
cf38 : a9 e0 85 fe a9 00 85 fd   3f
cf40 : 20 44 cf 60 78 a0 00 a2   54
cf48 : 32 a9 37 85 01 b1 fd 91   86
cf50 : fd c8 d0 f9 e6 fe ca d0   58
cf58 : f4 58 60 da ec 00 d9 ec   fc
cf60 : 00 86 02 05 35 e5 05 91   7e
cf68 : 02 80 1c e5 80 da e1 08   e5
cf70 : dc e1 01 d1 ec 16 d6 fd   8e
cf78 : e5 8f b7 05 8a 02 80 00   6e
cf80 : 00 93 ac ac ac 20 c4 41   6c
cf88 : 54 41 57 4f 52 4b 20 c2   c2
cf90 : 41 53 49 43 20 31 2e 31   dc
cf98 : 20 bb bb bb 0d 0d 28 c3   5d
cfa0 : 29 20 31 39 38 35 20 42   7f
cfa8 : 59 20 20 cb 4c 41 55 53   5e
cfb0 : 20 d3 54 52 55 4e 4b 0d   28
cfb8 : 0d 00 ff ff ff ff ff ff   c5
cfc0 : ff ff ff ff ff ff ff ff   bf
cfc8 : ff ff ff ff ff ff ff ff   c7
cfd0 : ff ff ff ff ff ff ff ff   cf
cfd8 : ff ff ff ff ff ff ff ff   d7
cfe0 : ff ff ff ff ff ff ff ff   df
cfe8 : ff ff ff ff ff ff ff ff   e7
cff0 : ff ff ff ff ff ff ff ff   ef
cff8 : ff ff ff ff ff ff ff ff   f7
```

Das Listing zum »Datawork-Basic« (Schluß)

Ascompiler 64

Ascompiler 64 ist ein kurzer und schnell arbeitender 3-Pass-Compiler, der ein vereinfachtes, sogenanntes »Tiny Basic« in reinen Maschinencode übersetzen kann. Dadurch lassen sich zeitkritische Routinen in Basic-Programmen, ohne Kenntnis von Assembler, erheblich beschleunigen.

Aus einem Quellprogramm, das unter Beachtung des eingeschränkten Befehlssatzes und der teilweise vereinfachten Befehlssyntax geschrieben ist, wird in Sekundenschnelle ein sehr effizienter Maschinencode erzeugt. Die Länge des erzeugten Codes entspricht in der Regel ungefähr der Länge des ursprünglichen Basic-Programms. Die Startadresse des Maschinenprogramms ist frei wählbar, die Ausführungsgeschwindigkeit liegt je nach konkretem Programm bis zu 200mal höher als beim Basic-Text. Im Durchschnitt laufen die mit dem Ascompiler 64 übersetzten Programme etwa 50- bis 120mal so schnell wie im normalen Basic. Dafür muß allerdings ein beschränkter Befehlssatz in Kauf genommen werden.

Der Ascompiler 64 belegt im Speicher den Bereich von $C000 bis $D000 und $E000 bis $F000; dazu kommen noch die Bereiche $A000 bis $B000 und $F000 bis $FFFF für Tabellen, die während des Compilierens angelegt werden. Die Compilierung eines Programms erfolgt direkt im Speicher und ist daher sehr schnell. Während Übersetzungszeiten von mehreren Stunden bei »großen« Basic-Compilern für den C 64 nicht ungewöhnlich sind, erfolgt die Übersetzung beim Ascompiler 64 bei kleineren Programmen (bis zu hundert Zeilen etwa) ohne merklichen Zeitverlust, bei längeren Programmen können es dann schon ein paar Sekunden werden.

Ein besonderer Vorteil dieses Compilers liegt in der Tatsache, daß die erzeugten Programme ganz einfach mit SYS aufgerufen werden können und nicht an bestimmte Speicheradressen gebunden sind. Es ist sogar möglich, bereits übersetzte Programmteile von einem weiteren Tiny-Basic-Programm aus per SYS aufzurufen. Da die Adressen der 26 möglichen Variablen (»A« bis »Z«) im Speicher festliegen (A: 828/829, B: 830/831,...Z: 878/879), ist es sogar möglich, Parameter von einem solchen Tiny-Basic-Programm an ein anderes zu übergeben.

Mit dem Ascompiler 64 ist es auch dem in der Maschinensprache nicht so bewanderten Basic-Programmierer endlich möglich, zeitkritische Unterroutinen ohne Problem schreiben zu können, um sie dann per SYS-Befehl vom eigentlichen Basic-Programm aus aufzurufen.

Wegen des Fehlens aller Fließkomma- und String-Operationen und wegen des insgesamt doch recht eingeschränkten Befehlssatzes ist der Ascompiler 64 nicht für größere Anwenderprogramme wie Dateiverwaltungen etc. gedacht. Ideale An-

wendungen dürften Grafik-Unterroutinen für hochauflösende Grafik und natürlich Action-Spiele aller Art sein — kurz, alle Programme, die im Wesentlichen aus PEEKs und POKEs und SYS-Aufrufen bestehen, sind für die Übersetzung mit dem Ascompiler 64 geradezu prädestiniert.

Der Umgang mit dem Compiler

Zunächst muß der Compiler natürlich in den Computer, was bei der allerersten Eingabe entweder durch Laden von der 64'er Leserservice-Diskette (geringster Aufwand) oder durch Eintippen von Listing 1 mittels MSE-Programm (geringste Kosten) möglich ist. Haben Sie sich für die zweite Methode entschieden, dann lesen Sie bitte zunächst die Eingabehinweise auf Seite 54, ehe Sie sich an die Arbeit machen.

Die in Listing 1 abgedruckte Version des Ascompilers liegt nicht im Original-Adreßbereich, sondern ist mit einem Basic-Header und einer nach »RUN« automatisch anlaufenden Verschieberoutine versehen. Wenn Sie das gesamte Listing mittels MSE richtig abgetippt und gespeichert haben, können Sie den Ascompiler 64 ganz einfach mit »LOAD« wie ein normales Basic-Programm laden. Nach »LIST« erscheint die Header-Zeile »10 SYS 2077 ASCOMPILER 64«.

Initialisiert wird der Ascompiler 64 mit »RUN«. Lassen Sie sich nicht irritieren, wenn danach der Bildschirm gelöscht wird und die READY-Meldung kommt. Der Ascompiler 64 befindet sich jetzt an der richtigen Adresse im Speicher und kann jederzeit mittels »SYS 49152« aktiviert werden. Das allerdings sollten Sie nur tun, wenn Sie ein entsprechendes Tiny-Basic-Programm im Speicher stehen haben.

Als ersten Test könnten Sie eingeben:

```
10 LET A=0
20 LET A=A+1
30 IF A< 60000 THEN 20
40 PRINT A
```

Lassen Sie dieses Programm einmal im normalen Basic laufen, um sich von der Richtigkeit zu überzeugen. Wenn Sie keine elf Minuten warten wollen, brechen Sie es einfach nach einer gewissen Zeit mit der STOP-Taste ab.

Rufen Sie jetzt den Compiler mit »SYS 49152« auf. Es erscheint eine Einschalt-Meldung des Compilers und die Abfrage auf die Startadresse des Maschinenprogramms, die hexadezimal angegeben werden muß. Wird ein unzulässiger Speicherbereich angegeben (zulässig ist alles vom Ende des Basic-Programms bis zum Ende des Basic-Speichers), so wird »NO MEMORY« ausgegeben und der Compilationsvorgang abgebrochen. Wird keine Adresse angegeben, dann wird das erzeugte Programm direkt hinter das Basic-Quellprogramm compiliert.

Ascompiler 64 ist ein Drei-Pass-Compiler, das heißt der Compiler führt intern drei verschiedene Durchläufe aus:

☐ Im ersten Durchgang (Pass 1) wird der Basic-Quelltext Befehl für Befehl direkt in Maschinencode umgesetzt. Dabei werden Texte aus PRINT-Anweisungen zunächst in einer vom Compiler angelegten Tabelle abgespeichert, von wo aus sie später an das fertige Compilat (das übersetzte Programm) angehängt werden. Außerdem legt der Compiler während dieses ersten Durchlaufs eine Tabelle der Zeilennummern und dazugehörigen effektiven Adressen an.

☐ In Pass 2 werden die Sprungadressen (GOTO, GOSUB und IF...THEN) im erzeugten Maschinenprogramm angepaßt, wobei auf die in Pass 1 erzeugte Zeilennummern-Tabelle zurückgegriffen wird.

☐ Im dritten Durchgang schließlich wird die in Pass 1 erzeugte Tabelle der PRINT-Texte an das Ende des compilierten Programms verschoben und die entsprechenden Zeiger innerhalb des Programms angepaßt.

Die Anfangs- und Endadresse des so erzeugten kompakten Maschinenprogramms wird nun vom Compiler in hexadezimaler Form ausgegeben und es erfolgt automatisch eine Rückkehr ins Basic. Das compilierte Programm kann nun mit SYS (Anfangsadresse) gestartet werden oder auch mit Hilfe eines Maschinensprache-Monitors abgespeichert werden. Natürlich muß zum Aufruf mit SYS die hexadezimal angegebene Startadresse in dezimale Form umgerechnet werden. Empfehlenswert ist daher die Wahl von »runden« hexadezimalen Startadressen: $1000 entspricht beispielsweise dezimal 4096 und ein Maschinenprogramm mit Startadresse $4000 kann mit »SYS 4*4096« gestartet werden. Zum Speichern ohne Monitor muß man zunächst die Anfangs- und die Endadresse in High- und Low-Byte aufteilen und diese dann als Dezimalwerte notieren. Mit der Befehlsfolge »POKE 43, (Start Low) : POKE 44, (Start High) : POKE 45, (Ende Low) : POKE 46, (Ende High) : SAVE "Name",8« kann das Maschinenprogramm auf Diskette gespeichert werden. Geladen wird es mit »LOAD"Name",8,1«. Datasettenbesitzer müssen die Geräteadresse 8 durch 1 ersetzen. Um weiter in Basic arbeiten zu können, muß der Befehl »NEW« sowohl nach dem Laden als auch nach dem Speichern gegeben werden, um die »verbogenen« Zeropage-Zeiger wieder zurechtzubiegen.

Ein zugegebenermaßen etwas umständliches Verfahren, aber das magere Commodore-Basic kann es nicht besser.

Eleganter geht es ohne Zweifel mit einem Maschinensprachemonitor, den aber wohl nicht jeder C 64-Besitzer zur Verfügung hat. In der nächsten Ausgabe bringen wir daher einen kleinen Mini-Monitor. Er ist natürlich in Tiny Basic geschrieben und mit dem Ascompiler 64 zu übersetzen. Damit wird es dann möglich sein, Maschinenprogramme mit einfachen Befehlen zu laden und zu speichern.

Tiny Basic und Commodore Basic

Programme, die vom Ascompiler übersetzt werden sollen, unterliegen gegenüber dem gewohnten Basic des C 64 einer Reihe von Einschränkungen:

— Pro Zeile darf nur ein einziger Befehl benutzt werden

— Zulässige Befehle sind LET, REM, CLR, END, GOTO, GOSUB, RETURN, SYS, PRINT, POKE, IF...THEN, STOP

— Der LET-Befehl ist nicht optional, sondern muß bei jeder Wertzuweisung an eine Variable ausgeschrieben werden

— In einem LET-Befehl darf nur eine Rechenoperation vorkommen. Kettenrechnungen oder Klammern sind nicht möglich. Beispielsweise muß die Zeile »10 LET A=4*X+1« ersetzt werden durch »10 LET A=4*X« und »11 LET A=A+1«

— In allen anderen Befehlen dürfen keine Rechenoperationen vorkommen. Es sind nur Variable oder positive Zahlen erlaubt.

— Erlaubte Rechenoperationen sind »+«, »-«, »*«, »/«, »AND« und »OR«

— Erlaubte Variablennamen sind nur die Buchstaben »A« bis »Z«. Alle Variablen sind ganzzahlig (16 Bit)

— Als Funktionen sind nur PEEK und RND vorhanden. Bei der RND-Funktion muß als Argument immer 255 angegeben werden. Es werden Zufallszahlen zwischen 0 und 255 geliefert. Beide Funktionen dürfen nicht in Rechnungen vorkommen. Beispiel: In »10 LET R=PEEK(X)+1« wird das »+1« vom Compiler ignoriert

— Hinter PRINT darf entweder ein Variablenname oder ein beliebiger Text in Anführungszeichen folgen. Rechnungen, Funktionen und Konstanten sind nicht zulässig

— In der IF-Anweisung sind die Vergleichsoperatoren »=«, »<« oder »>« zulässig. Der erste Vergleichsoperand muß eine Variable sein, die entweder mit einer anderen Variablen oder mit einem Zahlenwert verglichen werden kann

— Hinter THEN darf nur eine Zeilennummer folgen, weitere Befehle sind unzulässig

Der Ascompiler 64 kennt fünf Fehlermeldungen:

— NO-COMPILE ERROR: Ein Befehl konnte nicht übersetzt werden (tritt beispielsweise bei einem vergessenen LET auf)

— SPRUNG NACH UNDEF. ZEILE: Ein GOTO oder GOSUB zeigt auf eine nicht vorhandene Zeilennummer

— UNZUL. VARIABLE: Es wurde eine unzulässige Variable verwendet (der Compiler faßt auch Funktionen wie SIN etc. als unzulässige Variablen auf)

— SYNTAX ERROR: Allgemeiner Fehler im Aufbau einer Zeile

— NO-MEMORY ERROR: Der für Basic-Programme zulässige Speicherbereich wurde infolge einer falschen Speicherbereichsangabe überschritten.

Soviel zur Programmierung und Übersetzung von Tiny-Basic-Programmen für den Ascompiler 64.

Um Sie nach soviel Theorie noch ein bißchen auf den Geschmack am Ascompiler 64 zu bringen, zum Schluß noch ein paar Demo-Listings:

Listing 2 ist ein Beispiel für die Programmierung hochauflösender Grafik. Es handelt sich um einen einfachen Funktionenplotter für die hochauflösende Grafik. Die zu plottende Funktion wird in den Zeile 63000 bis 63020 definiert und in den für hochauflösende Grafik am C 64 zulässigen Wertebereich gebracht. Das Programm löscht zunächst das Farb-RAM, anschließend der HiRes Bit-Map. Wenn Sie viel Zeit haben, können Sie das Programm ja mal im normalen Basic in uncompilierter Form laufen lassen. In compilierter Form braucht es jedenfalls nur wenige Augenblicke ...

Ein weiteres eindrucksvolles Beispiel für die Geschwindigkeit des erzeugten Compilats ist in Listing 3 abgedruckt. Es handelt sich um eine Routine, die genau abgestimmt dann die Bildschirmfarben wechselt, wenn der Elektronenstrahl eine bestimmte Stelle am Bildschirm erreicht hat. Der Effekt: Der Bildschirm erscheint horizontal in zwei Farben geteilt.

Listing 4 schließlich stellt drei beim Ascompiler leider nicht vorhandene wichtige Funktionen in Form von kurzen Unterprogrammen zur Verfügung:

— GET Z (Aufruf mit GOSUB 10): Speichert den ASCII-Code der gerade gedrückten Taste in der Variablen Z. Wurde keine Taste gedrückt, enthält Z den Wert Null. Die Routine arbeitet über eine Abfrage des Tastaturpuffers

— PRINT CHR$(Z) (Aufruf mit GOSUB 20): Gibt das ASCII-Zeichen, das dem Wert von Z entspricht, auf dem Bildschirm aus. Diese Routine arbeitet mit einem speziellen Trick: Der Befehl »POKE 2,Z« in Zeile 20 wird vom Compiler nämlich übersetzt in die Befehlsfolge »LDA Z, STA 2«. Der Akku enthält also nach diesem Befehl noch das Low-Byte von Z, das durch Aufruf der Kernel-Routine BSOUT auf den Bildschirm kommt.

— INPUT(Z) (Aufruf mit GOSUB 30): Liest eine ganze Zeile von der Tastatur und legt sie in einem reservierten Speicherbereich ab. Die Anfangsadresse der Zeile wird in der Variablen Z an das aufrufende Programm zurückgegeben. Das Ende der Zeile ist durch ein Null-Byte markiert.

Damit sind zugleich einige Beispiele gegeben, wie man durch geschickte Ausnutzung von geeigneten Systemadressen auch eigentlich nicht vorgesehene Fähigkeiten wie eine einigermaßen komfortable Tastaturabfrage erreichen kann.

Zugegeben, die Programmierung für den Ascompiler 64 ist gewöhnungsbedürftig und im Vergleich zum normalen Basic umständlich und wenig komfortabel. Im Vergleich zur Programmierung in Assembler, die von der erreichbaren Geschwindigkeit her die einzige Alternative wäre, ist der Ascompiler 64 jedoch der reinste Luxus. Es liegt nur noch an Ihnen, ob Sie die dadurch gegebenen Möglichkeiten nutzen wollen. Sie haben jetzt auf jeden Fall ein System an der Hand, mit dem schnelle Action-Spiele und andere ehedem typische Assembler-Anwendungen in Basic programmiert werden können.

(Stephan Bartosch/ev)

```
@:ascomp programm : ascomp 0801 2871
------------------------------------

0801 : 1a 08 0a 00 9e 32 30 37  4d
0809 : 37 20 20 41 53 43 4f 4d  a8
0811 : 50 49 4c 45 52 20 36 34  29
0819 : 00 00 00 00 a9 70 85 5f  0c
0821 : a9 18 85 60 a9 72 85 5a  3d
0829 : a9 28 85 5b a9 02 85 58  25
0831 : a9 f0 85 59 20 bf a3 a9  c1
0839 : 6d 85 5f a9 08 85 60 a9  f7
0841 : 6f 85 5a a9 18 85 5b a9  ad
0849 : 02 85 58 a9 d0 85 59 20  38
0851 : bf a3 2f 0b d3 8d e4 92  71
0859 : d5 ed 30 f8 c5 6a d3 ce  ed
0861 : f6 6c 5c c4 eb a7 d0 2f  db
0869 : 8c 62 00 c0 a9 90 a0 cd  7c
0871 : 20 86 cc 20 cf ff c9 0d  4a
0879 : f0 35 20 7f c0 85 fe 8d  4b
0881 : 62 cc 8d 70 cc 20 7c c0  fc
0889 : 85 fd 8d 61 cc 8d 6f cc  2d
0891 : a5 2e cd 70 cc 90 2c d0  72
0899 : 07 a5 2d cd 6f cc 90 23  5e
08a1 : a9 37 85 01 58 a9 87 a0  9a
08a9 : ca 20 86 cc 4c 60 c5 a5  e9
08b1 : 2d 8d 6f cc 85 fd 8d 61  5b
08b9 : cc a5 2e 8d 70 cc 85 fe  17
08c1 : 8d 62 cc a9 31 a0 c9 20  67
08c9 : 86 cc 4c bb c0 8d 63 cc  df
08d1 : 4a 4a 4a 4a aa bd 5d c9  be
08d9 : 20 d2 ff ad 63 cc 29 0f  77
08e1 : aa bd 5d c9 20 d2 ff 60  54
08e9 : 20 cf ff c9 0d f0 32 a0  8c
08f1 : 00 d9 5d c9 f0 08 c0 0f  df
08f9 : f0 27 c8 4c 85 c0 98 0a  0d
0901 : 0a 0a 0a 8d 63 cc 20 cf  01
0909 : ff c9 0d f0 14 a0 00 d9  48
0911 : 5d c9 f0 08 c0 0f f0 09  ea
0919 : c8 4c a3 c0 98 0d 63 cc  21
0921 : 60 a2 f6 9a 4c 00 c0 78  9c
0929 : a9 35 85 01 a5 2b 85 fb  b0
0931 : a5 2c 85 fc a9 00 85 39  10
0939 : a9 a0 85 3a a9 e1 85 f9  8f
0941 : 8d 6b cc a9 e9 85 fa 8d  be
0949 : 6c cc d8 a2 f6 9a 20 50  0b
0951 : c1 48 20 ce c1 68 c9 8f  be
0959 : f0 f0 c9 80 d0 03 4c 04  a2
0961 : c2 c9 88 d0 03 4c ad ca  23
0969 : c9 89 d0 03 4c 5f c7 c9  fe
0971 : 8b d0 03 4c 87 c3 c9 8d  88
0979 : d0 03 4c 89 c7 c9 8e d0  b6
0981 : 03 4c 6e c2 c9 97 d0 03  41
0989 : 4c ec c7 c9 99 d0 03 4c  3b
0991 : 3c e7 c9 9c d0 03 4c 2b  73
0999 : c2 c9 9e d0 03 4c b4 c7  f6
09a1 : c9 90 f0 15 a9 37 85 01  fd
09a9 : 58 a9 6d a0 c9 20 86 cc  97
09b1 : ae 64 cc ad 65 cc 20 cd  53
09b9 : bd 4c 88 c1 a0 00 b1 fb  bf
09c1 : 8d 5f cc c8 b1 fb f0 2c  61
09c9 : 8d 60 cc c8 b1 fb 8d 64  cc
09d1 : cc c8 b1 fb 8d 65 cc c8  b6
09d9 : b1 fb 48 c8 98 18 65 fb  8b
09e1 : 85 69 a5 fc 69 00 85 6a  a5
09e9 : ae 5f cc 86 fb ae 60 cc  9b
09f1 : 86 fc 68 60 a9 60 a0 00  3c
09f9 : 91 fd c8 91 fd a5 fd 18  22
0a01 : 69 01 8d 6d cc a5 fe 69  c5
0a09 : 00 8d 6e cc a5 f9 8d 6b  3c
0a11 : cc a5 fa 8d 6c cc a2 f6  c6
0a19 : 9a a9 37 85 01 58 a9 4f  1e
0a21 : a0 ca 20 86 cc 20 98 c4  b9
0a29 : a9 6c a0 ca 20 86 cc 78  e4
0a31 : a9 35 85 01 20 09 e8 4c  7d
0a39 : d4 c6 a0 00 ad 64 cc 91  ed
0a41 : f9 c8 ad 65 cc 91 f9 c8  89
0a49 : a5 fd 91 f9 c8 a5 fe 91  69
0a51 : f9 c8 98 18 65 f9 85 f9  08
0a59 : a5 fa 69 00 85 fa a5 fa  92
0a61 : c9 ff 90 0b d0 06 a5 f9  77
0a69 : c9 f0 90 03 4c 34 c0 60  59
0a71 : a9 18 a0 c2 a2 1d 8e 67  c3
0a79 : cc a2 c2 8e 68 cc 20 3f  05
0a81 : c2 4c de c0 a2 80 6c 00  19
0a89 : 03 00 a0 ff c8 a9 00 99  c1
0a91 : 3c 03 c0 33 d0 f6 00 a9  fd
0a99 : 1e a0 c2 a2 2a 8e 67 cc  5b
0aa1 : a2 c2 8e 68 cc 20 3f c2  a5
0aa9 : 4c de c0 85 6f 84 70 a5  6d
0ab1 : fd a4 fe 20 af c6 a0 00  78
0ab9 : b1 6f 91 fd a5 6f cd 67  22
0ac1 : cc d0 08 a5 70 cd 68 cc  5d
0ac9 : d0 01 60 e6 6f d0 02 e6  62
0ad1 : 70 e6 fd d0 da e6 fe 4c  c7
0ad9 : 43 c2 a9 82 a0 c2 a2 83  ea
0ae1 : 8e 67 cc a2 c2 8e 68 cc  86
0ae9 : 20 3f c2 4c de c0 60 00  58
0af1 : 8c 66 cc a0 00 b1 69 48  bb
0af9 : a5 69 18 69 01 85 69 a5  b3
0b01 : 6a 69 00 85 6a 68 c9 20  22
0b09 : f0 e9 8d 83 cc ad 83 cc  a3
0b11 : c9 3a b0 06 38 e9 30 38  e8
0b19 : e9 d0 ac 66 cc 60 a0 00  b5
0b21 : b1 69 e6 69 d0 02 e6 6a  fb
0b29 : 60 20 17 c7 ad 71 cc 8d  0d
0b31 : 11 c3 ad 72 cc 8d 12 c3  e6
0b39 : ad 69 cc 8d 15 c3 ad 75  91
0b41 : cc 8d 17 c3 ad 76 cc 8d  ef
0b49 : 18 c3 ad 73 cc 8d 1a c3  46
0b51 : ad 74 cc 8d 1b c3 ad 6a  78
0b59 : cc 8d 1d c3 ad 77 cc 8d  90
0b61 : 1f c3 ad 78 cc 8d 20 c3  1d
0b69 : a9 10 a0 c3 a2 21 8e 67  f7
0b71 : cc a2 c3 8e 68 cc 20 3f  3d
0b79 : c2 4c de c0 ad 3c 03 18  2a
0b81 : 69 01 8d 3c 03 ad 3d 03  ee
0b89 : 69 00 8d 3d 03 00 20 17  dc
0b91 : c7 ad 71 cc 8d 76 c3 ad  1c
0b99 : 72 cc 8d 77 c3 ad 69 cc  ac
0ba1 : 8d 7a c3 ad 75 cc 8d 7c  ff
0ba9 : c3 ad 76 cc 8d 7d c3 ad  a9
0bb1 : 73 cc 8d 7f c3 ad 74 cc  f3
0bb9 : 8d 80 c3 ad 6a cc 8d 82  75
0bc1 : c3 ad 77 cc 8d 84 c3 ad  3a
0bc9 : 78 cc 8d 85 c3 a9 75 a0  5c
```

Listing 1. »Ascompiler 64«. Bitte mit dem MSE eingeben.

```
0bd1 : c3 a2 86 8e 67 cc a2 c3   48
0bd9 : 8e 68 cc 20 3f c2 4c de   cb
0be1 : c0 ad ff ff 38 e9 00 8d   66
0be9 : ff ff ad ff ff e9 00 8d   bd
0bf1 : ff ff 00 20 84 c2 20 e8   a5
0bf9 : c8 20 84 c2 c9 b1 d0 03   be
0c01 : 4c 9d cc c9 b2 d0 03 4c   de
0c09 : 08 cf c9 b3 f0 03 4c 47   c8
0c11 : c8 20 84 c2 b0 03 4c 2b   0d
0c19 : c4 ad 71 cc 8d 20 c4 ad   f2
0c21 : 72 cc 8d 21 c4 ad 73 cc   a2
0c29 : 8d 14 c4 ad 74 cc 8d 15   b5
0c31 : c4 20 a1 c2 20 e8 c8 ad   8e
0c39 : 71 cc 8d 23 c4 ad 72 cc   f5
0c41 : 8d 24 c4 ad 73 cc 8d 17   c9
0c49 : c4 ad 74 cc 8d 18 c4 20   87
0c51 : 84 c2 c9 a7 f0 03 4c 47   84
0c59 : c8 20 84 c2 20 17 c7 ad   e0
0c61 : 69 cc 8d 28 c4 ad 6a cc   96
0c69 : 8d 29 c4 a9 13 a0 c4 a2   80
0c71 : 2a 8e 67 cc a2 c4 8e 68   b1
0c79 : cc 20 3f c2 4c de c0 ad   98
0c81 : ff ff cd ff ff f0 04 b0   ec
0c89 : 0d 90 08 ad ff ff cd ff   cd
0c91 : ff b0 03 4c ff ff 00 20   73
0c99 : 17 c7 ad 71 cc 8d 8e c4   2a
0ca1 : ad 72 cc 8d 8f c4 ad 73   29
0ca9 : cc 8d 83 c4 ad 74 cc 8d   82
0cb1 : 84 c4 ad 69 cc 8d 91 c4   39
0cb9 : ad 6a cc 8d 86 c4 20 a1   d2
0cc1 : c2 c9 a7 f0 03 4c 47 c8   b1
0cc9 : 20 84 c2 20 17 c7 ad 69   19
0cd1 : cc 8d 95 c4 ad 6a cc 8d   de
0cd9 : 96 c4 a9 82 a0 c4 a2 97   76
0ce1 : 8e 67 cc a2 c4 8e 68 cc   a6
0ce9 : 20 3f c2 4c de c0 ad 3d   08
0cf1 : 03 c9 01 f0 04 b0 0c 90   4e
0cf9 : 07 ad 3c 03 c9 40 b0 03   ae
0d01 : 4c 1e 00 00 78 a9 35 85   11
0d09 : 01 ad 6f cc 85 fd ad 70   36
0d11 : cc 85 fe a0 00 b1 fd c9   8d
0d19 : 4c f0 52 c9 20 f0 4e a0   af
0d21 : 00 b1 fd 20 e0 c4 a5 fe   46
0d29 : cd 6e cc d0 0a a5 fd cd   dc
0d31 : 6d cc d0 03 4c da c4 8a   5d
0d39 : 18 65 fd 85 fd a5 fe 69   10
0d41 : 00 85 fe 4c a7 c4 a9 37   03
0d49 : 85 01 58 60 a2 01 c9 20   0b
0d51 : f0 18 29 9f f0 16 29 1f   2e
0d59 : c9 09 f0 0f c9 19 f0 0a   02
0d61 : 29 0d c9 08 f0 06 29 08   78
0d69 : f0 01 e8 e8 60 a0 03 b1   ab
0d71 : fd c9 ea f0 aa a9 e1 85   b6
0d79 : f9 a9 e9 85 fa a0 01 b1   8e
0d81 : fd a0 00 d1 f9 d0 1d a0   e4
0d89 : 02 b1 fd a0 01 d1 f9 d0   1f
0d91 : 13 a0 02 b1 f9 a0 01 91   77
0d99 : fd a0 03 b1 f9 a0 02 91   ad
0da1 : fd 4c b3 c4 a5 fa cd 6c   8c
0da9 : cc f0 02 90 29 a5 f9 cd   c3
0db1 : 6b cc 90 22 a9 37 85 01   57
0db9 : 58 a9 d6 a0 c9 20 86 cc   01
0dc1 : a0 01 b1 fd aa a0 02 b1   29
0dc9 : fd 20 cd bd a9 f7 a0 c9   72
0dd1 : 20 86 cc 4c 7a c5 a5 f9   51
0dd9 : 18 69 04 85 f9 a5 fa 69   e3
0de1 : 00 85 fa 4c 11 c5 a2 f6   a4
0de9 : 9a a9 37 85 01 58 a2 80   35
0df1 : 6c 00 03 20 84 c2 b0 03   49
0df9 : 4c 95 c5 20 e8 c8 4c c0   0d
0e01 : c2 20 17 c7 ad 69 cc 8d   1c
0e09 : 6d 79 cc 8d 69 cc ad 6a   a0
0e11 : cc 6d 7a cc 8d 6a cc 4c   c4
0e19 : a5 c8 20 84 c2 b0 03 4c   11
0e21 : 22 c3 ad 71 cc 8d 19 c6   e9
0e29 : ad 72 cc 8d 1a c6 ad 73   6a
0e31 : cc 8d 23 c6 ad 74 cc 8d   32
0e39 : 24 c6 20 a1 c2 20 e8 c8   5f
0e41 : ad 71 cc 8d 1d c6 ad 72   2f
0e49 : cc 8d 1e c6 ad 73 cc 8d   01
0e51 : 26 c6 ad 74 cc 8d 27 c6   38
0e59 : ad 75 cc 8d 20 c6 ad 76   81
0e61 : cc 8d 21 c6 ad 77 cc 8d   fa
0e69 : 29 c6 ad 78 cc 8d 2a c6   df
0e71 : a9 18 a0 c6 a2 2b 8e 67   b4
0e79 : cc a2 c6 8e 68 cc 20 3f   06
0e81 : c2 4c de c0 ad ff ff 38   84
0e89 : ed ff ff 8d ff ff ad ff   de
0e91 : ff ed ff ff 8d ff ff 00   5f
0e99 : 20 84 c2 b0 03 4c 4d c6   17
0ea1 : 20 17 c7 ad 79 cc 38 ed   af
0ea9 : 69 cc 8d 69 cc ad 7a cc   c7
0eb1 : ed 6a cc 8d 69 cc 4c a5   32
0eb9 : c8 ad 69 cc 8d 9e c6 ad   90
0ec1 : 71 cc 8d a1 c6 ad 72 cc   6d
0ec9 : 8d a2 c6 ad 75 cc 8d a4   4c
0ed1 : c6 ad 76 cc 8d a5 c6 ad   21
0ed9 : 6a cc 8d a7 c6 ad 73 cc   43
0ee1 : 8d a9 c6 ad 74 cc 8d aa   e3
0ee9 : c6 ad 77 cc 8d ac c6 ad   b2
0ef1 : 78 cc 8d ad c6 a9 9d a0   5a
0ef9 : c6 a2 ae 8e 67 cc a2 c6   83
0f01 : 8e 68 cc 20 3f c2 4c de   f3
0f09 : c0 a9 00 38 ed ff ff 8d   9f
0f11 : ff ff a9 00 ed ff ff 8d   74
0f19 : ff ff 00 c4 38 90 20 d0   db
0f21 : 04 c5 37 90 1a 48 98 48   bf
0f29 : a9 37 85 01 58 a9 a2 a0   8e
0f31 : ca 20 86 cc 68 20 61 c0   d5
0f39 : 68 20 61 c0 4c 34 c0 60   4c
0f41 : a9 37 85 01 58 a9 8f a0   5a
0f49 : c9 20 86 cc ae 64 cc ad   fa
0f51 : 65 cc 20 cd bd a9 bd a0   3f
0f59 : c9 20 86 cc ad 70 cc 20   3f
0f61 : 61 c0 ad 6f cc 20 61 c0   50
0f69 : a9 cf a0 c9 20 86 cc ad   20
0f71 : 6e cc 20 61 c0 ad 6d cc   42
0f79 : 20 61 c0 a9 0d 20 d2 ff   cc
0f81 : 4c 7a c5 a2 00 8e 69 cc   84
0f89 : 8e 6a cc b0 39 e9 2f 85   40
0f91 : 07 ad 6a cc 85 22 ad 69   96
0f99 : cc 0a 26 22 0a 26 22 6d   6d
0fa1 : 69 cc 8d 69 cc a5 22 6d   5e
0fa9 : 6a cc 8d 6a cc 0e 69 cc   a6
0fb1 : 2e 6a cc ad 69 cc 65 07   9e
0fb9 : 8d 69 cc 90 03 ee 6a cc   2b
0fc1 : 20 84 c2 4c 1f c7 60 4c   a8
0fc9 : 00 00 00 20 84 c2 20 17   da
0fd1 : c7 ad 69 cc 8d 5c c7 ad   99
0fd9 : 6a cc 8d 5d c7 a9 5b a0   31
0fe1 : c7 a2 5e 8e 67 cc a2 c7   5a
0fe9 : 8e 68 cc 20 3f c2 4c de   db
0ff1 : c0 20 00 00 00 20 84 c2   5a
0ff9 : 20 17 c7 ad 69 cc 8d 86   8d
1001 : c7 ad 6a cc 8d 87 c7 a9   5a
1009 : 85 a0 c7 a2 88 8e 67 cc   59
1011 : a2 c7 8e 68 cc 20 3f c2   98
1019 : 4c de c0 20 00 00 ea 00   b4
1021 : 20 84 C2 20 17 C7 AD 69   71
1029 : CC 8D B0 C7 AD 6A CC 8D   5D
1031 : B1 C7 A9 AF A0 C7 A2 B3   60
1039 : 8E 67 CC A2 C7 8E 68 CC   2E
1041 : 20 3F C2 4C DE C0 AE FF   EA
1049 : FF AD FF FF 86 FE 85 FF   95
1051 : A0 00 AD FF FF 91 FE [illegible]   E5
1059 : 20 84 C2 B0 03 4C 64 E9   7A
1061 : 20 E8 C8 20 84 C2 C9 2C   09
1069 : F0 03 4C 47 C8 20 84 C2   FC
1071 : B0 03 4C 1E E9 AD 71 CC   E5
1079 : 8D DB C7 AD 72 CC 8D DC   19
1081 : C7 AD 73 CC 8D DE C7 AD   DF
1089 : 74 CC 8D DF C7 20 A1 C2   4C
1091 : 20 E8 C8 AD 71 CC 8D E7   91
1099 : C7 AD 72 CC 8D E8 C7 A9   FF
10A1 : DA A0 C7 A2 EB 8E 67 CC   7C
10A9 : A2 C7 8E 68 CC 20 3F C2   30
10B1 : 4C DE C0 A9 37 85 01 58   26
10B9 : A9 33 A0 CA 20 86 CC AE   44
10C1 : 64 CC AD 65 CC 20 CD BD   24
10C9 : 4C 60 C5 20 17 C7 AD 69   F4
10D1 : CC 8D 79 CC AD 6A CC 8D   D8
10D9 : 7A CC 20 A1 C2 C9 00 D0   12
10E1 : 03 4C A5 C8 C9 AA D0 03   C8
10E9 : 4C 87 C5 C9 AB D0 03 4C   89
10F1 : 2C C6 C9 AC D0 03 4C 3F   5D
10F9 : E3 C9 AD D0 03 4C 39 E6   8C
1101 : C9 AF D0 03 4C 33 E1 C9   B0
1109 : B0 D0 03 4C 56 E1 4C 47   A0
1111 : C8 AD 69 CC 8D DE C8 AD   F2
1119 : 75 CC 8D E0 C8 AD 76 CC   E1
1121 : 8D E1 C8 AD 6A CC 8D E3   92
1129 : C8 AD 77 CC 8D E5 C8 AD   C6
1131 : 78 CC 8D E6 C8 A9 DD A0   E2
1139 : C8 A2 E7 8E 67 CC A2 C8   17
1141 : 8E 68 CC 20 3F C2 4C DE   33
1149 : C0 A9 01 8D 3C 03 A9 00   52
1151 : 8D 3D 03 00 38 E9 41 0A   2A
1159 : 18 69 3C 8D 71 CC A9 03   11
1161 : 69 00 8D 72 CC AD 72 CC   1A
1169 : C9 03 D0 07 AD 71 CC C9   F6
1171 : 70 90 18 A9 37 85 01 58   B9
1179 : A9 17 A0 CA 20 86 CC AD   F4
1181 : 65 CC AE 64 CC 20 CD BD   05
1189 : 4C 60 C5 AD 71 CC 18 69   DD
1191 : 01 8D 73 CC AD 72 CC 69   44
1199 : 00 8D 74 CC 60 0D 0D 11   DB
11A1 : 20 2D 2D 2D 2D 2D 2D 2D   94
11A9 : 2D 2D 2D 2D 20 C2 C9 D4   47
11B1 : D4 C5 20 D7 C1 D2 D4 C5   FC
11B9 : CE 20 2D 2D 2D 2D 2D 2D   D4
11C1 : 2D 2D 2D 2D 2D 2D 2D 0D   81
11C9 : 00 30 31 32 33 34 35 36   8A
11D1 : 37 38 39 41 42 43 44 45   75
11D9 : 46 0D 0D 2A 2A 2A 20 4E   40
11E1 : 4F 20 43 4F 4D 50 49 4C   10
11E9 : 45 20 45 52 52 4F 52 20   03
11F1 : 2A 2A 2A 20 20 20 49 4E   84
11F9 : 20 20 00 0D 20 C3 4F 4D   C3
1201 : 50 49 4C 49 45 52 55 4E   0B
1209 : 47 20 42 45 45 4E 44 45   FC
1211 : 54 0D 20 C3 4F 4D 50 49   A0
1219 : 4C 49 45 52 54 20 42 49   87
1221 : 53 20 DA 45 49 4C 45 20   30
1229 : 00 0D 0D 20 CD C3 2D D0   48
1231 : 52 47 4D 2E 20 56 4F 4E   CF
1239 : 20 24 00 20 42 49 53 20   6b
1241 : 24 00 0d 20 d3 50 52 55   60
1249 : 4e 47 20 4e 41 43 48 20   9c
1251 : 55 4e 44 45 46 2e 0d 20   d1
1259 : da 45 49 4c 45 20 ce 52   e7
1261 : 2e 20 00 0d 0d 20 20 c3   1b
1269 : cf cd d0 c9 cc c9 c5 d2   64
1271 : d5 ce c7 a0 c1 c2 c7 c5   90
1279 : c2 d2 cf c3 c8 c5 ce a0   48
1281 : 21 0d 00 0d 20 d5 4e 5a   69
1289 : 55 4c 2e 20 d6 41 52 49   e7
1291 : 41 42 4c 45 20 49 4e 20   75
1299 : da 45 49 4c 45 20 00 0d   61
12a1 : 2a 2a 2a 20 53 59 4e 54   51
12a9 : 41 58 20 45 52 52 4f 52   61
12b1 : 20 2a 2a 2a 20 20 49 4e   7b
12b9 : 20 20 00 0d 20 20 d0 41   54
12c1 : 53 53 20 32 20 2d 20 d3   a0
12c9 : 50 52 55 4e 47 41 44 44   7a
12d1 : 52 45 53 53 45 4e 0d 00   00
12d9 : 0d 20 20 d0 41 53 53 20   55
12e1 : 33 20 2d 20 d0 d2 c9 ce   dc
12e9 : d4 20 d4 41 42 45 4c 4c   43
12f1 : 45 0d 00 0d 0d 2a 2a 2a   7e
12f9 : 20 ce 4f 2d cd 45 4d 4f   d5
1301 : 52 59 20 c5 52 52 4f 52   5a
1309 : 20 2a 2a 2a 0d 00 0d 0d   2d
1311 : 20 20 2d 20 24 00 4c 47   93
1319 : c8 20 84 c2 20 e8 c8 ad   32
1321 : 71 cc 8d 75 cc ad 72 cc   a8
1329 : 8d 76 cc ad 73 cc 8d 77   9d
1331 : cc ad 74 cc 8d 78 cc 20   9b
1339 : 84 c2 c9 b2 d0 d8 20 84   44
1341 : c2 b0 03 4c 5f c8 c9 c2   8f
1349 : d0 03 4c f1 cb c9 bb d0   88
1351 : 03 4c 00 e0 20 e8 c8 20   43
1359 : 84 c2 c9 00 d0 03 4c a0   48
1361 : cb c9 aa d0 03 4c 22 cb   88
1369 : c9 ab d0 03 4c ae c5 c9   81
1371 : ac d0 03 4c 57 e2 c9 ad   df
1379 : d0 03 4c c2 e4 c9 af d0   33
1381 : 03 4c 52 e0 c9 b0 d0 03   c6
1389 : 4c 79 e1 4c 47 c8 20 84   d8
1391 : c2 b0 03 4c bd c2 ad 71   81
1399 : cc 8d 8d cb ad 72 cc 8d   c5
13a1 : 8e cb ad 73 cc 8d 97 cb   1e
13a9 : ad 74 cc 8d 98 cb 20 a1   21
13b1 : c2 20 e8 c8 ad 71 cc 8d   8b
13b9 : 91 cb ad 72 cc 8d 92 cb   05
13c1 : ad 73 cc 8d 9a cb ad 74   b4
13c9 : cc 8d 9b cb ad 75 cc 8d   91
13d1 : 94 cb ad 76 cc 8d 95 cb   ac
13d9 : ad 77 cc 8d 9d cb ad 78   06
13e1 : cc 8d 9e cb a9 8c a0 cb   ae
13e9 : a2 9f 8e 67 cc a2 cb 8e   19
13f1 : 68 cc 20 3f c2 4c de c0   3b
13f9 : ad ff ff 18 6d ff ff 8d   9b
1401 : ff ff ad ff ff 6d ff ff   d6
1409 : 8d ff ff 00 ad 71 cc 8d   4a
1411 : e5 cb ad 72 cc 8d e6 cb   02
1419 : ad 75 cc 8d e8 cb ad 76   f6
1421 : cc 8d e9 cb ad 73 cc 8d   6c
1429 : eb cb ad 74 cc 8d ec cb   78
1431 : ad 77 cc 8d ee cb ad 78   73
1439 : cc 8d ef cb a9 e4 a0 cb   1d
1441 : a2 f0 8e 67 cc a2 cb 8e   1a
1449 : 68 cc 20 3f c2 4c de c0   93
1451 : ad ff ff 8d ff ff ad ff   66
1459 : ff 8d ff ff 00 20 84 c2   b7
1461 : c9 28 f0 03 4c 47 c8 20   3d
1469 : 84 c2 b0 03 4c d7 e8 20   42
1471 : e8 c8 ad 71 cc 8d 4b cc   57
1479 : ad 72 cc 8d 4c cc ad 73   0d
1481 : cc 8d 50 cc ad 74 cc 8d   8e
1489 : 51 cc ad 75 cc 8d 5c cc   9f
1491 : ad 76 cc 8d 5d cc ad 77   40
1499 : cc 8d 57 cc ad 78 cc 8d   88
14a1 : 58 cc a9 4a a0 cc a2 5e   cb
14a9 : 8e 67 cc a2 cc 8e 68 cc   ef
14b1 : 20 3f c2 4c de c0 ad ff   56
14b9 : ff 85 fe ad ff ff 85 ff   06
```

Listing 1. »Ascompiler 64« (Fortsetzung)



```
14c1 : a0 00 8c ff ff b1 fe 8d  29
14c9 : ff ff 00 00 08 00 60 c3  52
14d1 : 3c 00 04 97 c4 1e 00 04  47
14d9 : a0 c3 60 00 60 3c 03 3d  e1
14e1 : 03 3c 03 3d 03 01 00 00  a3
14e9 : 00 00 00 00 00 00 00 00  ea
14f1 : 00 00 85 6b 84 6c a0 00  ee
14f9 : b1 6b f0 0c 20 d2 ff e6  84
1501 : 6b d0 f3 e6 6c 4c 8a cc  9b
1509 : 60 20 84 c2 b0 03 4c 23  8d
1511 : cd ad 71 cc 8d 1b cd ad  ef
1519 : 72 cc 8d 1c cd ad 73 cc  8a
1521 : 8d 0f cd ad 74 cc 8d 10  63
1529 : cd 20 a1 c2 20 e8 c8 ad  8f
1531 : 71 cc 8d 18 cd ad 72 cc  1c
1539 : 8d 19 cd ad 73 cc 8d 0c  68
1541 : cd ad 74 cc 8d 0d cd 20  54
1549 : 84 c2 c9 a7 f0 03 4c 47  7c
1551 : c8 20 84 c2 20 17 c7 ad  d8
1559 : 69 cc 8d 20 cd ad 6a cc  1d
1561 : 8d 21 cd a9 0b a0 cd a2  5a
1569 : 22 8e 67 cc a2 cd 8e 68  e9
1571 : cc 20 3f c2 4c de c0 ad  90
1579 : ff ff cd ff ff f0 04 90  a4
1581 : 0a b0 0b ad ff ff cd ff  93
1589 : ff b0 03 4c ff ff 00 20  6b
1591 : 17 c7 ad 71 cc 8d 88 cd  1c
1599 : ad 72 cc 8d 89 cd ad 73  09
15a1 : cc 8d 7d cd ad 74 cc 8d  1a
15a9 : 7e cd ad 69 cc 8d 86 cd  95
15b1 : ad 6a cc 8d 7b cd 20 a1  62
15b9 : c2 c9 a7 f0 03 4c 47 c8  a9
15c1 : 20 84 c2 20 17 c7 ad 69  11
15c9 : cc 8d 8d cd ad 6a cc 8d  f5
15d1 : 8e cd a9 7a a0 cd a2 8f  22
15d9 : 8e 67 cc a2 cd 8e 68 cc  2f
15e1 : 20 3f c2 4c de c0 a9 00  76
15e9 : cd ff ff f0 04 90 09 b0  1e
15f1 : 0a a9 00 cd ff ff b0 03  52
15f9 : 4c ff ff 00 0e 08 93 11  d6
1601 : 11 11 11 12 1d 1d 1d 1d  8b
1609 : 20 20 20 20 20 20 20 20  09
1611 : 20 20 20 20 20 20 20 20  11
1619 : 20 20 20 20 20 20 20 20  19
1621 : 20 20 20 20 20 20 20 20  21
1629 : 0d 12 1d 1d 1d 1d 20 20  a6
1631 : 20 c1 a0 d3 a0 c3 a0 cf  1f
1639 : a0 cd a0 d0 a0 c9 a0 cc  76
1641 : a0 c5 a0 d2 20 2d 20 36  9f
1649 : 20 34 20 20 20 20 0d 12  eb
1651 : 1d 1d 1d 1d 20 20 20 20  ac
1659 : 20 20 20 20 20 20 20 20  59
1661 : 20 20 20 20 20 20 20 20  61
1669 : 20 20 20 20 20 20 20 20  69
1671 : 20 20 20 20 0d 0d 1d 1d  95
1679 : c4 41 4e 4b 45 20 41 4e  d2
1681 : 20 3a 20 c7 45 52 44 20  f8
1689 : cb 41 55 54 5a 4d 41 4e  86
1691 : 4e 20 46 55 45 52 20 53  3a
1699 : 45 49 4e 45 0d 1d 1d 20  2e
16a1 : 20 20 20 20 20 20 20 20  a1
16a9 : 20 20 d5 45 42 45 52 45  19
16b1 : 44 55 4e 47 53 4b 55 4e  9e
16b9 : 53 54 20 55 4e 44 0d 0d  3e
16c1 : 1d 1d 20 20 20 20 20 20  3d
16c9 : 20 20 20 20 20 d0 52 4f  76
16d1 : 46 49 2d c1 53 53 20 36  fc
16d9 : 34 20 2e 0d 0d 1d 1d 1d  b3
16e1 : 1d 20 28 43 29 20 31 39  4c
16e9 : 38 34 20 42 59 20 53 54  18
16f1 : 45 50 48 41 4e 20 42 41  0a
16f9 : 52 54 4f 53 43 48 0d 1d  98
1701 : 1d 1d 1d 20 20 20 20 20  bc
1709 : 20 20 20 20 20 20 20 20  09
1711 : 41 4c 54 52 48 45 49 4e  48
1719 : 53 54 52 2e 20 32 35 20  99
1721 : 0d 1d 1d 1d 1d 20 20 20  3b
1729 : 20 20 20 20 20 20 20 20  29
1731 : 20 20 36 38 34 33 20 c2  d9
1739 : c9 c2 cc c9 d3 20 2d 20  03
1741 : 32 20 0d 0d 1d 1d c1 4e  c7
1749 : 46 41 4e 47 53 41 44 52  a1
1751 : 45 53 53 45 20 44 45 53  9d
1759 : 20 0d 1d 1d cd 41 53 43  a6
1761 : 48 49 4e 45 4e 2d d0 52  c0
1769 : 4f 47 52 41 4d 4d 53 20  e5
1771 : 3a 20 24 00 20 84 c2 b0  57
1779 : 03 4c 8a cf ad 71 cc 8d  f4
1781 : 7f cf ad 72 cc 8d 80 cf  7c
1789 : ad 73 cc 8d 77 cf ad 74  6a
1791 : cc 8d 78 cf 20 a1 c2 20  96
1799 : e8 c8 ad 71 cc 8d 82 cf  62
17a1 : ad 72 cc 8d 83 cf ad 73  c0
17a9 : cc 8d 7a cf ad 74 cc 8d  a1
17b1 : 7b cf 20 84 c2 c9 a7 f0  a7
17b9 : 03 4c 47 c8 20 84 c2 20  3f
```

```
17c1 : 17 c7 ad 69 cc 8d 87 cf  4b
17c9 : ad 6a cc 8d 88 cf a9 76  2b
17d1 : a0 cf a2 89 8e 67 cc a2  cf
17d9 : cf 8e 68 cc 20 3f c2 4c  43
17e1 : de c0 ad ff ff cd ff ff  f9
17e9 : d0 0b ad ff ff cd ff ff  18
17f1 : d0 03 4c ff ff 00 20 17  04
17f9 : c7 ad 71 cc 8d e9 cf ad  4f
1801 : 72 cc 8d ea cf ad 73 cc  6c
1809 : 8d e2 cf ad 74 cc 8d e3  5c
1811 : cf ad 69 cc 8d ec cf ad  86
1819 : 6a cc 8d e5 cf 20 a1 c2  13
1821 : c9 a7 f0 03 4c 47 c8 20  bd
1829 : 84 c2 20 17 c7 ad 69 cc  22
1831 : 8d f0 cf ad 6a cc 8d f1  07
1839 : cf a9 e1 a0 cf a2 f2 8e  64
1841 : 67 cc a2 cf 8e 68 cc 20  50
1849 : 3f c2 4c de c0 ad ff ff  52
1851 : c9 00 d0 0a ad ff ff c9  fe
1859 : 00 d0 03 4c ff ff 00 00  0c
1861 : 00 00 ff 00 00 00 00 f5  4d
1869 : ff 00 ff ff ff 00 e0 20  2c
1871 : 84 c2 c9 28 f0 03 4c 47  b5
1879 : c8 20 84 c2 20 17 c7 ad  00
1881 : 69 cc c9 ff d0 f0 ad 75  f9
1889 : cc 8d 4a e0 ad 76 cc 8d  a7
1891 : 4b e0 ad 77 cc 8d 4f e0  df
1899 : ad 78 cc 8d 50 e0 a9 43  a0
18a1 : a0 e0 a2 51 8e 67 cc a2  21
18a9 : e0 8e 68 cc 20 3f c2 4c  24
18b1 : de c0 ae 04 dc bd 00 f0  b9
18b9 : 8d ff ff a9 00 8d ff ff  e7
18c1 : 00 20 84 c2 b0 03 4c cf  3f
18c9 : e0 ad 71 cc 8d bd e0 ad  1b
18d1 : 72 cc 8d be e0 ad 73 cc  c7
18d9 : 8d c6 e0 ad 74 cc 8d c7  2b
18e1 : e0 20 a1 c2 20 e8 c8 ad  5a
18e9 : 71 cc 8d c0 e0 ad 72 cc  1b
18f1 : 8d c1 e0 ad 73 cc 8d c9  b4
18f9 : e0 ad 74 cc 8d ca e0 ad  74
1901 : 75 cc 8d c3 e0 ad 76 cc  a7
1909 : 8d c4 e0 ad 77 cc 8d cc  94
1911 : e0 ad 78 cc 8d cd e0 a9  9e
1919 : bc a0 e0 a2 ce 8e 67 cc  4a
1921 : a2 e0 8e 68 cc 20 3f c2  34
1929 : 4c de c0 ad ff ff 2d ff  7f
1931 : ff 8d ff ff ad ff ff 2d  2c
1939 : ff ff 8d ff ff 00 20 17  4a
1941 : c7 ad 71 cc 8d 23 e1 ad  aa
1949 : 72 cc 8d 24 e1 ad 73 cc  fc
1951 : 8d 2b e1 ad 74 cc 8d 2c  de
1959 : e1 ad 69 cc 8d 26 e1 ad  f2
1961 : 6a cc 8d 2e e1 ad 75 cc  55
1969 : 8d 28 e1 ad 76 cc 8d 29  8f
1971 : e1 ad 77 cc 8d 30 e1 ad  dd
1979 : 78 cc 8d 31 e1 a9 22 a0  16
1981 : e1 a2 32 8e 67 cc a2 e1  3d
1989 : 8e 68 cc 20 3f c2 4c de  7b
1991 : c0 ad ff ff 29 00 8d ff  f0
1999 : ff ad ff ff 29 00 8d ff  37
19a1 : ff 00 20 84 c2 90 06 20  42
19a9 : e8 c8 4c d2 e0 20 17 c7  5e
19b1 : ad 69 cc 2d 79 cc 8d 69  f3
19b9 : cc ad 6a cc 2d 7a cc 8d  85
19c1 : 6a cc 4c a5 c8 20 84 c2  7e
19c9 : b0 18 20 17 c7 ad 69 cc  99
19d1 : 0d 79 cc 8d 69 cc ad 6a  08
19d9 : cc 0d 7a cc 8d 6a cc 4c  5c
19e1 : a5 c8 20 e8 c8 4c 81 e1  c8
19e9 : 20 84 c2 b0 64 20 17 c7  45
19f1 : ad 71 cc 8d d2 e1 ad 72  13
19f9 : cc 8d d3 e1 ad 69 cc 8d  31
1a01 : d5 e1 ad 75 cc 8d d7 e1  3d
1a09 : ad 76 cc 8d d8 e1 ad 73  10
1a11 : cc 8d da e1 ad 74 cc 8d  63
1a19 : db e1 ad 6a cc 8d dd e1  12
1a21 : ad 77 cc 8d df e1 ad 78  23
1a29 : cc 8d e0 e1 a9 d1 a0 e1  9f
1a31 : a2 e1 8e 67 cc a2 e1 8e  db
1a39 : 68 cc 20 3f c2 4c de c0  83
1a41 : ad ff ff 09 00 8d ff ff  7b
1a49 : ad ff ff 09 00 8d ff ff  83
1a51 : 00 ad 71 cc 8d 45 e2 ad  08
1a59 : 72 cc 8d 46 e2 ad 73 cc  60
1a61 : 8d 4e e2 ad 74 cc 8d 4f  06
1a69 : e2 20 a1 c2 20 e8 c8 ad  e4
1a71 : 71 cc 8d 48 e2 ad 72 cc  b4
1a79 : 8d 49 e2 ad 73 cc 8d 51  90
1a81 : e2 ad 74 cc 8d 52 e2 ad  43
1a89 : 75 cc 8d 4b e2 ad 76 cc  40
1a91 : 8d 4c e2 ad 77 cc 8d 54  6f
1a99 : e2 ad 78 cc 8d 55 e2 a9  6c
1aa1 : 44 a0 e2 a2 56 8e 67 cc  53
1aa9 : a2 e2 8e 68 cc 20 3f c2  bd
1ab1 : 4c de c0 ad ff ff 0d ff  86
1ab9 : ff 8d ff ff ad ff ff 0d  73
```

```
1ac1 : ff ff 8d ff ff 00 20 84  ad
1ac9 : c2 b0 03 4c 39 e3 ad 71  7a
1ad1 : cc 8d e3 e2 8d 1e e3 ad  6e
1ad9 : 72 cc 8d e4 e2 8d 1f e3  90
1ae1 : ad 73 cc 8d 24 e3 ad 74  2e
1ae9 : cc 8d 25 e3 20 a1 c2 20  9c
1af1 : e8 c8 ad 71 cc 8d e9 e2  7d
1af9 : 8d 18 e3 ad 72 cc 8d ea  da
1b01 : e2 8d 19 e3 ad 73 cc 8d  31
1b09 : de e2 8d ef e2 ad 74 cc  c1
1b11 : 8d df e2 8d f0 e2 ad 75  c0
1b19 : cc 8d 30 e3 ad 76 cc 8d  11
1b21 : 31 e3 ad 77 cc 8d 36 e3  78
1b29 : ad 78 cc 8d 37 e3 a9 d3  d8
1b31 : a0 e2 a2 38 8e 67 cc a2  8f
1b39 : e3 8e 68 cc 20 3f c2 4c  b7
1b41 : de c0 a2 08 a9 00 8d fa  f0
1b49 : 03 8d fb 03 ad ff ff f0  2f
1b51 : 35 ad ff ff 8d f0 03 ad  24
1b59 : ff ff 8d f5 03 ad ff ff  17
1b61 : 8d f6 03 4e f0 03 90 13  83
1b69 : 18 ad fa 03 6d f5 03 8d  24
1b71 : fa 03 ad fb 03 6d f6 03  55
1b79 : 8d fb 03 0e f5 03 2e f6  a5
1b81 : 03 ca d0 df f0 15 ad ff  88
1b89 : ff 8d f0 03 ad ff ff 8d  e1
1b91 : f5 03 ad ff ff 8d f6 03  c1
1b99 : 38 b0 c8 ad fa 03 8d ff  0f
1ba1 : ff ad fb 03 8d ff ff 00  af
1ba9 : 20 17 c7 4c 4a e3 20 84  1e
1bb1 : c2 b0 03 4c 82 e4 20 e8  b7
1bb9 : c8 ad 6a cc f0 03 4c e5  b0
1bc1 : e3 ad 69 cc 8d a7 e3 ad  70
1bc9 : 71 cc 8d ac e3 ad 72 cc  a8
1bd1 : 8d ad e3 ad 73 cc 8d b2  1d
1bd9 : e3 ad 74 cc 8d b3 e3 ad  ab
1be1 : 75 cc 8d dc e3 ad 76 cc  da
1be9 : 8d dd e3 ad 77 cc 8d e2  ed
1bf1 : e3 ad 78 cc 8d e3 e3 a9  3d
1bf9 : 9c a0 e3 a2 e4 8e 67 cc  2c
1c01 : a2 e3 8e 68 cc 20 3f c2  96
1c09 : 4c de c0 a2 08 a9 00 8d  32
1c11 : fa 03 8d fb 03 a9 00 8d  08
1c19 : f0 03 ad ff ff 8d f5 03  40
1c21 : ad ff ff 8d f6 03 4e f0  22
1c29 : 03 90 13 18 ad fa 03 6d  d6
1c31 : f5 03 8d fa 03 ad fb 03  fe
1c39 : 6d f6 03 8d fb 03 0e f5  90
1c41 : 03 2e f6 03 ca d0 df ad  87
1c49 : fa 03 8d ff ff ad fb 03  8b
1c51 : 8d ff ff 00 ad 71 cc 8d  92
1c59 : 45 e4 ad 72 cc 8d 46 e4  e6
1c61 : ad 73 cc 8d 36 e4 ad 74  d7
1c69 : cc 8d 37 e4 ad 69 cc 8d  db
1c71 : 4b e4 ad 6a cc 8d 50 e4  2b
1c79 : ad 75 cc 8d 79 e4 ad 76  28
1c81 : cc 8d 7a e4 ad 77 cc 8d  34
1c89 : 7f e4 ad 78 cc 8d 80 e4  fa
1c91 : a9 35 a0 e4 a2 81 8e 67  d9
1c99 : cc a2 e4 8e 68 cc 20 3f  ad
1ca1 : c2 4c de c0 ad ff ff d0  d5
1ca9 : 47 a2 08 a9 00 8d fa 03  d7
1cb1 : 8d fb 03 ad ff ff 8d f0  ca
1cb9 : 03 a9 00 8d f5 03 a9 00  61
1cc1 : 8d f6 03 4e f0 03 90 13  e3
1cc9 : 18 ad fa 03 6d f5 03 8d  84
1cd1 : fa 03 ad fb 03 6d f6 03  b5
1cd9 : 8d fb 03 0e f5 03 2e f6  05
1ce1 : 03 ca d0 df ad fa 03 8d  53
1ce9 : ff ff ad fb 03 8d ff ff  6f
1cf1 : 00 20 17 c7 a2 10 a9 00  11
1cf9 : 8d 84 cc 8d 85 cc 4e 6a  7a
1d01 : cc 6e 69 cc 90 13 18 ad  56
1d09 : 84 cc 6d 79 cc 8d 84 cc  63
1d11 : ad 85 cc 6d 7a cc 8d 85  b1
1d19 : cc 0e 79 cc 2e 7a cc ca  64
1d21 : d0 dc ad 84 cc 8d 69 cc  d4
1d29 : ad 85 cc 8d 6a cc 4c a5  07
1d31 : c8 20 84 c2 b0 03 4c 8a  ec
1d39 : e5 ad 71 cc 8d 37 e5 ad  70
1d41 : 72 cc 8d 38 e5 ad 73 cc  b7
1d49 : 8d 3d e5 ad 74 cc 8d 3e  04
1d51 : e5 20 a1 c2 20 e8 c8 ad  cf
1d59 : 71 cc 8d 43 e5 ad 72 cc  2b
1d61 : 8d 44 e5 ad 73 cc 8d 49  a6
1d69 : e5 ad 74 cc 8d 4a e5 ad  fa
1d71 : 77 cc 8d 84 e5 ad 78 cc  89
1d79 : 8d 85 e5 ad 75 cc 8d 81  ef
1d81 : e5 ad 76 cc 8d 82 e5 a9  4c
1d89 : 2c a0 e5 a2 89 8e 67 cc  17
1d91 : a2 e5 8e 68 cc 20 3f c2  27
1d99 : 4c de c0 a2 10 a9 00 8d  42
1da1 : f8 03 8d f9 03 ad 3e 03  5a
1da9 : 8d f4 03 ad 3f 03 8d f5  55
1db1 : 03 ad 3c 03 8d f6 03 ad  f2
1db9 : 3d 03 8d f7 03 0e f4 03  55
```

```
1dc1 : 2e f5 03 2e f8 03 2e f9  c5
1dc9 : 03 ad f9 03 cd f7 03 90  4b
1dd1 : 1e d0 08 ad f8 03 cd f6  dc
1dd9 : 03 90 14 38 ad f8 03 ed  bb
1de1 : f6 03 8d f8 03 ad f9 03  67
1de9 : ed f7 03 8d f9 03 38 2e  39
1df1 : 40 03 2e 41 03 ca d0 c5  bc
1df9 : 00 20 17 c7 ad 71 cc 8d  7d
1e01 : e8 e5 ad 72 cc 8d e9 e5  42
1e09 : ad 73 cc 8d ee e5 ad 74  12
1e11 : cc 8d ef e5 ad 69 cc 8d  d1
1e19 : f4 e5 ad 6a cc 8d f9 e5  a5
1e21 : ad 75 cc 8d 30 e6 ad 76  4b
1e29 : cc 8d 31 e6 ad 77 cc 8d  ca
1e31 : 33 e6 ad 78 cc 8d 34 e6  2a
1e39 : a9 dd a0 e5 a2 38 8e 67  ab
1e41 : cc a2 e6 8e 68 cc 20 3f  d6
1e49 : c2 4c de c0 a2 10 a9 00  52
1e51 : 8d f8 03 8d f9 03 ad 3c  b4
1e59 : 03 8d f4 03 ad 3d 03 8d  ac
1e61 : f5 03 a9 02 8d f6 03 a9  72
1e69 : 00 8d f7 03 0e f4 03 2e  7f
1e71 : f5 03 2e f8 03 2e f9 03  22
1e79 : ad f9 03 cd f7 03 90 1e  b3
1e81 : d0 08 ad f8 03 cd f6 03  60
1e89 : 90 14 38 ad f8 03 ed f6  34
1e91 : 03 8d f8 03 ad f9 03 ed  8c
1e99 : f7 03 8d f9 03 38 2e 40  e0
1ea1 : 03 2e 41 03 ca d0 c5 00  b6
1ea9 : 20 84 c2 b0 03 4c f0 e6  f6
1eb1 : 20 e8 c8 ad 69 cc 8d 9f  a0
1eb9 : e6 ad 6a cc 8d a4 e6 ad  9f
1ec1 : 71 cc 8d a9 e6 ad 72 cc  70
1ec9 : 8d aa e6 ad 73 cc 8d af  4e
1ed1 : e6 ad 74 cc 8d b0 e6 ad  9a
1ed9 : 75 cc 8d e7 e6 ad 76 cc  64
1ee1 : 8d e8 e6 ad 77 cc 8d ea  3b
1ee9 : e6 ad 78 cc 8d eb e6 a9  85
1ef1 : 94 a0 e6 a2 ef 8e 67 cc  8e
1ef9 : a2 e6 8e 68 cc 20 3f c2  0f
1f01 : 4c de c0 a2 10 a9 00 8d  aa
1f09 : f8 03 8d f9 03 a9 2c 8d  6f
1f11 : f4 03 a9 01 8d f5 03 ad  01
1f19 : 3c 03 8d f6 03 ad 3d 03  b2
1f21 : 8d f7 03 0e f4 03 2e f5  38
1f29 : 03 2e f8 03 2e f9 03 ad  fc
1f31 : f9 03 cd f7 03 90 1e d0  ed
1f39 : 08 ad f8 03 cd f6 03 90  78
1f41 : 14 38 ad f8 03 ed f6 03  7d
1f49 : 8d f8 03 ad f9 03 ed f7  28
1f51 : 03 8d f9 03 38 2e 40 03  f6
1f59 : 2e 41 03 ca d0 c5 00 20  bd
1f61 : 17 c7 a9 00 8d 84 cc 8d  11
1f69 : 85 cc ad 79 cc 38 ed 69  08
1f71 : cc 8d 79 cc 90 16 ad 7a  61
1f79 : cc ed 6a cc 8d 7a cc 90  71
1f81 : 18 ee 84 cc d0 e4 ee 85  c6
1f89 : cc 4c fb e6 08 ad 7a cc  c8
1f91 : d0 02 f0 04 28 4c 07 e7  f0
1f99 : 28 ad 84 cc 8d 69 cc ad  05
1fa1 : 85 cc 8d 6a cc 4c a5 c8  94
1fa9 : 43 7c 60 20 84 c2 c9 00  cc
1fb1 : f0 4d 8d 39 e7 c9 22 f0  0a
1fb9 : 61 c9 3b f0 ee 20 e8 c8  11
1fc1 : ad 71 cc 8d 84 e7 ad 72  2f
1fc9 : cc 8d 85 e7 ad 73 cc 8d  7f
1fd1 : 87 e7 ad 74 cc 8d 88 e7  71
1fd9 : a9 7d a0 e7 a2 8f 8e 67  15
1fe1 : cc a2 e7 8e 68 cc 20 3f  b6
1fe9 : c2 4c 3c e7 a9 20 20 d2  9f
1ff1 : ff ea ae 40 03 ad 41 03  c2
1ff9 : 20 cd bd ea e6 d3 00 ad  35
2001 : 39 e7 c9 3b f0 11 a9 02  4a
2009 : a0 e8 a2 08 8e 67 cc a2  64
2011 : e8 8e 68 cc 20 3f c2 4c  94
2019 : de c0 20 b2 c2 c9 22 f0  9b
2021 : 28 c9 00 f0 1e a6 3a e0  0e
2029 : bf 90 0b d0 06 a6 39 e0  49
2031 : ff 90 03 4c 34 c0 a0 00  8f
2039 : 91 39 e6 39 d0 dc e6 3a  4c
2041 : 4c ab e7 4c 47 c8 4c de  90
2049 : c0 a9 00 a8 91 39 e6 39  e4
2051 : d0 02 e6 3a a9 f9 a0 e7  e0
2059 : a2 01 8e 67 cc a2 e8 8e  af
2061 : 68 cc 20 3f c2 4c 3c e7  6f
2069 : a9 00 a0 00 20 1e ab ea  b2
2071 : 00 a9 0d 20 d2 ff ea 00  66
2079 : ad 6d cc 8d 3a e7 85 f9  ae
2081 : ad 6e cc 8d 3b e7 85 fa  49
2089 : a5 39 8d 6b cc a5 3a 8d  9a
2091 : 6c cc a9 00 85 39 a9 a0  d8
2099 : 85 3a a5 f9 a4 fa 20 af  e6
20a1 : c6 a0 00 b1 39 91 f9 ad  51
20a9 : 6c cc c5 3a d0 0a ad 6b  1f
20b1 : cc c5 39 d0 03 4c 58 e8  8e
20b9 : e6 39 d0 02 e6 3a e6 f9  80
20c1 : d0 d8 e6 fa 4c 2b e8 ad  33
20c9 : 6d cc 85 39 ad 6e cc 85  b1
20d1 : 3a a5 f9 18 69 01 8d 6d  0f
20d9 : cc a5 fa 69 00 8d 6e cc  23
20e1 : ad 6f cc 85 fd ad 70 cc  d2
20e9 : 85 fe a0 00 b1 fd c9 a9  9b
20f1 : f0 1f a0 00 b1 fd 20 e0  e6
20f9 : c4 ad 3b e7 c5 fe d0 08  07
2101 : ad 3a e7 c5 fd d0 01 60  a9
2109 : e6 fd d0 de e6 fe 4c 7b  8c
2111 : e8 a0 02 b1 fd c9 a0 d0  52
2119 : d9 a0 04 b1 fd c9 20 d0  ca
2121 : d1 a0 05 b1 fd c9 1e d0  02
2129 : c9 a0 01 a5 39 91 fd a0  91
2131 : 03 a5 3a 91 fd a0 00 b1  10
2139 : 39 e6 39 d0 02 e6 3a c9  22
2141 : 00 d0 f2 4c 83 e8 20 17  1e
2149 : c7 ad 75 cc 8d 1b e9 ad  92
2151 : 76 cc 8d 1c e9 ad 77 cc  98
2159 : 8d 15 e9 ad 78 cc 8d 16  f1
2161 : e9 ad 69 cc 8d 18 e9 ad  b1
2169 : 6a cc 8d 19 e9 a9 12 a0  35
2171 : e9 a2 1d 8e 67 cc a2 e9  00
2179 : 8e 68 cc 20 3f c2 4c de  6b
2181 : c0 a9 00 8d ff ff ad ff  7e
2189 : ff 8d ff ff 00 20 17 c7  3c
2191 : ad 71 cc 8d 56 e9 ad 72  2c
2199 : cc 8d 57 e9 ad 73 cc 8d  04
21a1 : 59 e9 ad 74 cc 8d 5a e9  5f
21a9 : ad 69 cc 8d 60 e9 a9 53  92
21b1 : a0 e9 a2 63 8e 67 cc a2  f7
21b9 : e9 8e 68 cc 20 3f c2 4c  3d
21c1 : de c0 a0 00 ad ff ff ae  60
21c9 : ff ff 85 fe 86 ff a9 00  18
21d1 : 91 fe 00 20 17 c7 20 a1  59
21d9 : c2 c9 2c f0 03 4c 47 c8  ea
21e1 : 20 84 c2 b0 03 4c af e9  2f
21e9 : 20 e8 c8 ad 71 cc 8d a9  6c
21f1 : e9 ad 72 cc 8d aa e9 ad  18
21f9 : 69 cc 8d ac e9 ad 6a cc  10
2201 : 8d ad e9 a9 a8 a0 e9 a2  91
2209 : ae 8e 67 cc a2 e9 8e 68  f6
2211 : cc 20 3f c2 4c de c0 ad  30
2219 : ff ff 8d ff ff 00 ad 69  05
2221 : cc 8d de e9 ad 6a cc 8d  25
2229 : df e9 20 a1 c2 20 17 c7  52
2231 : ad 69 cc 8d dc e9 a9 db  f3
2239 : a0 e9 a2 e0 8e 67 cc a2  2f
2241 : e9 8e 68 cc 20 3f c2 4c  c5
2249 : de c0 a9 37 8d 01 00 00  b9
2251 : 0a 00 01 60 14 00 06 60  c2
2259 : 1e 00 10 60 28 00 1a 60  33
2261 : 32 00 2e 60 3c 00 3f 60  ac
2269 : 46 00 50 60 50 00 61 60  1b
2271 : 5a 00 76 60 ff ff ff ff  75
2279 : ff ff ff ff ff ff ff ff  78
2281 : ff ff ff ff ff ff ff ff  80
2289 : ef ff 00 ff ff ff ff fb  70
2291 : ff ff ff ff ff ff ff ff  90
2299 : ff ff ff ff ff ff ff ff  98
22a1 : ff ff ff ff ff ff ff ff  a0
22a9 : ff ff ff ff 6b ff 00 ff  5e
22b1 : d5 00 00 ef 00 ff ff ff  84
22b9 : 00 00 00 ba ff ff ff ff  11
22c1 : 00 00 00 45 c1 00 e5 00  1e
22c9 : 00 00 ff c0 00 ff ff 00  e1
22d1 : 00 00 00 00 00 ff 00 00  d1
22d9 : 00 00 df 00 00 ff ff 00  d1
22e1 : 00 00 00 00 00 ff 00 00  e1
22e9 : 00 00 f5 ff 00 ff ff 00  67
22f1 : ff ff ff ff ff ff ff ff  f0
22f9 : ff ff ff ff ff ff ff ff  f8
2301 : ff ff ff ff ff ff ff ff  00
2309 : ef ff 00 ff ff ff ff fb  f0
2311 : ff ff ff ff ff ff ff ff  10
2319 : ff ff ff ff ff ff ff ff  18
2321 : ff ff ff ff ff ff ff ff  20
2329 : ff ff ff ff 6b ff 00 ff  de
2331 : d5 00 00 ff 00 ff ff ff  06
2339 : 00 00 00 ba ff ff ff ff  91
2341 : 00 00 00 45 d1 00 e5 00  9f
2349 : 00 00 ff c0 00 ff ff 00  61
2351 : 00 00 00 00 00 ff 00 00  51
2359 : 00 00 df 00 00 ff ff 00  51
2361 : 00 00 00 00 00 ff 00 00  61
2369 : 00 00 f5 ff 00 ff ff 04  ef
2371 : ff ff ff ff ff ff ff ff  70
2379 : ff ff ff ff ff ff ff ff  78
2381 : ff ff ff ff ff ff ff ff  80
2389 : ef ff 00 ff ff ff ff fb  70
2391 : ff ff ff ff ff ff ff ff  90
2399 : ff ff ff ff ff ff ff ff  98
23a1 : ff ff ff ff ff ff ff ff  a0
23a9 : ff ff ff ff 6b ff 00 ff  5e
23b1 : d5 00 00 ff 00 ff ff ff  86
23b9 : 00 00 00 ba ff ff ff ff  11
23c1 : 00 00 00 45 d1 00 e5 00  1f
23c9 : 00 00 ff c0 00 ff ff 00  e1
23d1 : 00 00 00 00 00 ff 00 00  d1
23d9 : 00 00 df 00 00 ff ff 00  d1
23e1 : 00 00 00 00 00 ff 00 00  e1
23e9 : 00 00 f5 ff 00 ff ff 00  67
23f1 : ff ff ff ff ff ff ff ff  f0
23f9 : ff ff ff ff ff ff ff ff  f8
2401 : ff ff ff ff ff ff ff ff  00
2409 : ef ff 00 ff ff ff ff fb  f0
2411 : ff ff ff ff ff ff ff ff  10
2419 : ff ff ff ff ff ff ff ff  18
2421 : ff ff ff ff ff ff ff ff  20
2429 : ff ff ff ff 6b ff 00 ff  de
2431 : d5 00 00 ff 00 ff ff ff  06
2439 : 00 00 00 ba ff ff ff ff  91
2441 : 00 00 00 45 c1 00 e5 00  9e
2449 : 00 00 ff c0 00 ff ff 00  61
2451 : 00 00 00 00 00 ff 00 00  51
2459 : 00 00 df 00 00 ff ff 00  51
2461 : 00 00 00 00 00 ff 00 00  61
2469 : 00 00 f5 ff 00 ff ff 04  ef
2471 : ff ff ff ff ff ff ff ff  70
2479 : ff ff ff ff ff ff ff ff  78
2481 : ff ff ff ff ff ff ff ff  80
2489 : ef ff 00 ff ff ff ff fb  70
2491 : ff ff ff ff ff ff ff ff  90
2499 : ff ff ff ff ff ff ff ff  98
24a1 : ff ff ff ff ff ff ff ff  a0
24a9 : ff ff ff ff 6b ff 00 ff  5e
24b1 : d5 00 00 ef 00 ff ff ff  84
24b9 : 00 00 00 ba ff ff ff ff  11
24c1 : 00 00 00 45 c1 00 e5 00  1e
24c9 : 00 00 ff c0 00 ff ff 00  e1
24d1 : 00 00 00 00 00 ff 00 00  d1
24d9 : 00 00 df 00 00 ff ff 00  d1
24e1 : 00 00 00 00 00 ff 00 00  e1
24e9 : 00 00 f5 ff 00 ff ff 00  67
24f1 : ff ff ff ff ff ff ff ff  f0
24f9 : ff ff ff ff ff ff ff ff  f8
2501 : ff ff ff ff ff ff ff ff  00
2509 : ef ff 00 ff ff ff ff fb  f0
2511 : ff ff ff ff ff ff ff ff  10
2519 : ff ff ff ff ff ff ff ff  18
2521 : ff ff ff ff ff ff ff ff  20
2529 : ff ff ff ff 6b ff 00 ff  de
2531 : d5 00 00 ff 00 ff ff ff  06
2539 : 00 00 00 ba ff ff ff ff  91
2541 : 00 00 00 45 c1 00 e5 00  9e
2549 : 00 00 ff c0 00 ff ff 00  61
2551 : 00 00 00 00 00 ff 00 00  51
2559 : 00 00 df 00 00 ff ff 00  51
2561 : 00 00 00 00 00 ff 00 00  61
2569 : 00 00 f5 ff 00 ff ff 04  ef
2571 : ff ff ff ff ff ff ff ff  70
2579 : ff ff ff ff ff ff ff ff  78
2581 : ff ff ff ff ff ff ff ff  80
2589 : ef ff 00 ff ff ff ff fb  70
2591 : ff ff ff ff ff ff ff ff  90
2599 : ff ff ff ff ff ff ff ff  98
25a1 : ff ff ff ff ff ff ff ff  a0
25a9 : ff ff ff ff 6b ff 00 ff  5e
25b1 : d5 00 00 ff 00 ff ff ff  86
25b9 : 00 00 00 ba ff ff ff ff  11
25c1 : 00 00 00 45 c1 00 e5 00  1e
25c9 : 00 00 ff c0 00 ff ff 00  e1
25d1 : 00 00 00 00 00 ff 00 00  d1
25d9 : 00 00 df 00 00 ff ff 00  d1
25e1 : 00 00 00 00 00 ff 00 00  e1
25e9 : 00 00 f5 ff 00 ff ff 00  67
25f1 : ff ff ff ff ff ff ff ff  f0
25f9 : ff ff ff ff ff ff ff ff  f8
2601 : ff ff ff ff ff ff ff ff  00
2609 : ef ff 00 ff ff ff ff fb  f0
2611 : ff ff ff ff ff ff ff ff  10
2619 : ff ff ff ff ff ff ff ff  18
2621 : ff ff ff ff ff ff ff ff  20
2629 : ff ff ff ff 6b ff 00 ff  de
2631 : d5 00 00 ef 00 ff ff ff  04
2639 : 00 00 00 ba ff ff ff ff  91
2641 : 00 00 00 45 c1 00 e5 00  9e
2649 : 00 00 ff c0 00 ff ff 00  61
2651 : 00 00 00 00 00 ff 00 00  51
2659 : 00 00 df 00 00 ff ff 00  51
2661 : 00 00 00 00 00 ff 00 00  61
2669 : 00 00 f5 ff 00 ff ff 04  ef
2671 : ff ff ff ff ff ff ff ff  70
2679 : ff ff ff ff ff ff ff ff  78
2681 : ff ff ff ff ff ff ff ff  80
2689 : ef ff 00 ff ff ff ff fb  70
2691 : ff ff ff ff ff ff ff ff  90
2699 : ff ff ff ff ff ff ff ff  98
26a1 : ff ff ff ff ff ff ff ff  a0
```

Listing 1. »Ascompiler 64« (Fortsetzung)

```
26a9 : ff ff ff ff 6b ff 00 ff  5e
26b1 : d5 00 00 ff 00 ff ff ff  86
26b9 : 00 00 00 ba ff ff ff ff  11
26c1 : 00 00 00 45 c1 00 e5 00  1e
26c9 : 00 00 ff c0 00 ff ff 00  e1
26d1 : 00 00 00 00 00 ff 00 00  d1
26d9 : 00 00 df 00 00 ff ff 00  d1
26e1 : 00 00 00 00 00 ff 00 00  e1
26e9 : 00 00 f5 ff 00 ff ff 00  67
26f1 : ff ff ff ff ff ff ff ff  f0
26f9 : ff ff ff ff ff ff ff ff  f8
2701 : ff ff ff ff ff ff ff ff  00
2709 : ef ff 00 ff ff ff ff fb  f0
2711 : ff ff ff ff ff ff ff ff  10
2719 : ff ff ff ff ff ff ff ff  18
2721 : ff ff ff ff ff ff ff ff  20
2729 : ff ff ff ff 6b ff 00 ff  de
2731 : d5 00 00 ff 00 ff ff ff  06
2739 : 00 00 00 ba ff ff ff ff  91
2741 : 00 00 00 45 c1 00 e5 00  9e
2749 : 00 00 ff c0 00 ff ff 00  61
2751 : 00 00 00 00 00 ff 00 00  51
2759 : 00 00 df 00 00 ff ff 00  51
2761 : 00 00 00 00 00 ff 00 00  61
2769 : 00 00 f5 ff 00 ff ff 04  ef
2771 : ff ff ff ff ff ff ff ff  70
2779 : ff ff ff ff ff ff ff ff  78
2781 : ff ff ff ff ff ff ff ff  80
2789 : ef ff 00 ff ff ff ff fb  70
2791 : ff ff ff ff ff ff ff ff  90
2799 : ff ff ff ff ff ff ff ff  98
27a1 : ff ff ff ff ff ff ff ff  a0
27a9 : ff ff ff ff 6b ff 00 ff  5e
27b1 : d5 00 00 ef 00 ff ff ff  84
27b9 : 00 00 00 ba ff ff ff ff  11
27c1 : 00 00 00 45 c1 00 e5 00  1e
27c9 : 00 00 ff c0 00 ff ff 00  e1
27d1 : 00 00 00 00 00 ff 00 00  d1
27d9 : 00 00 df 00 00 ff ff 00  d1
27e1 : 00 00 00 00 00 ff 00 00  e1
27e9 : 00 00 f5 ff 00 ff ff 00  67
27f1 : ff ff ff ff ff ff ff ff  f0
27f9 : ff ff ff ff ff ff ff ff  f8
2801 : ff ff ff ff ff ff ff ff  00
2809 : ef ff 00 ff ff ff ff fb  f0
2811 : ff ff ff ff ff ff ff ff  10
2819 : ff ff ff ff ff ff ff ff  18
2821 : ff ff ff ff ff ff ff ff  20
2829 : ff ff ff ff 6b ff 00 ff  de
2831 : d5 00 00 ff 00 ff ff ff  06
2839 : 00 00 00 ba ff ff ff ff  91
2841 : 00 00 00 45 c1 00 e5 00  9e
2849 : 00 00 ff c0 00 ff ff 00  61
2851 : 00 00 00 00 00 ff 00 00  51
2859 : 00 00 df 00 00 ff ff 00  51
2861 : 00 00 00 00 00 ff 00 00  61
2869 : 00 00 f5 ff 00 ff ff ff  e7
```

Listing 1. »Ascompiler 64« (Schluß)

```
10 REM  HIRES-PLOT-DEMO
20 REM FUER ASCOMPILER 64
30 REM
60000 POKE 53265,59
60010 POKE 53272,24
60011 POKE 20000,128
60012 POKE 20001,64
60013 POKE 20002,32
60014 POKE 20003,16
60015 POKE 20004,8
60016 POKE 20005,4
60017 POKE 20006,2
60018 POKE 20007,1
60020 LET I=1024
60030 POKE I,1
60040 LET I=I+1
60050 IF I<2023 THEN 60030
60060 LET I=8192
60070 POKEI,0
60075 LET I=I+1
60080 IF I<16384 THEN 60070
60090 GOTO 63000
61000 LET W=Y/8
61010 LET V=X/8
61020 LET H=W*40
61030 LET H=H+V
61040 LET H=H*8
61050 LET H=H+8192
61060 LET G=W*8
61070 LET G=Y-G
61080 LET P=H+G
61100 IF P>16384 THEN 63900
61110 LET B=X/8
61120 LET B=B*8
61130 LET B=X-B
61140 LET Q=20000+B
61150 LET Q=PEEK(Q)
61155 LET L=PEEK(P)
61156 LET L=LORQ
61160 POKE P,L
61170 RETURN
63000 LET X=0
63005 LET Y=X*X
63010 LET Y=Y+X
63020 LET Y=Y/200
63030 GOSUB 61000
63040 LET X=X+1
63050 IF X<400 THEN 63005
63900 LET Q=PEEK(203)
63910 IF Q=64 THEN 63900
63940 POKE 53265,27
63950 POKE 53272,23
63960 PRINT"□";
63970 END

READY.
```

Listing 2. Ein Mini-Plot-Programm für den Ascompiler 64

```
1 REM  SPLITSCREEN-DEMO
2 REM FUER ASCOMPILER 64
3 REM
10 LET A=PEEK(53266)
20 IF A>0 THEN 10
30 POKE 53281,2
40 LET A=PEEK(53266)
50 IF A<128 THEN 40
60 POKE 53281,3
70 LET A=PEEK(197)
80 IF A=64 THEN 10

READY.
```

Listing 3. Dieses »Screen-Split«-Demo zeigt die mit dem Ascompiler erreichbaren Geschwindigkeiten

```
1 REM DREI NUETZLICHE ROUTINEN
2 REM FUER DEN ASCOMPILER 64
3 REM
7 REM
8 REM GET Z
9 REM
10 LET Z=PEEK(198)
11 IF Z=0 THEN 14
12 LET Z=PEEK(631)
13 POKE 198,0
14 RETURN
17 REM
18 REM PRINT CHR$(Z)
19 REM
20 POKE 2,Z
21 SYS 65490
22 RETURN
27 REM
28 REM INPUT(Z)
29 REM
30 SYS 42336
31 LET Z=512
32 RETURN

READY.
```

Listing 4. Drei nützliche Unterroutinen

Hardcopy leicht gemacht

Jetzt bekommt Ihr Drucker was zu tun! Wir zeigen Ihnen, wie Sie die Basic-Hardcopy-Routine aus der Ausgabe 9/85 in Maschinensprache umsetzen können. Ein gedrucktes »Foto« des Textbildschirms auf Tastendruck.

Eine Hardcopy vom Textbildschirm des C 64 wird häufig gebraucht. Die Dokumentation eines selbstgeschriebenen Programms oder ein Beispielausdruck für eine Programmanleitung sind nur zwei Beispiele.

In der letzten Folge war die Hardcopy-Routine auf Commodore-Drucker zugeschnitten. Diesmal sind alle 8-Nadel-Drukker an der Reihe. Das Programm ist auf die Epson-Drucker FX und RX zugeschnitten. Ein verwendetes Interface muß sich durch die Sekundäradresse 1 im OPEN-Befehl in den Linearmodus schalten lassen. Lassen Sie sich davon aber nicht abschrecken, falls Ihr Interface oder Drucker auf andere Weise angesteuert wird. Wenn Sie Grundkenntnisse in der Assembler-Programmierung haben, sind Sie in der Lage, das Programm auch an Ihren Drucker anzupassen. Voraussetzung für das Gelingen der Anpassung ist nur, daß Ihr Drucker über einen Bitmustermodus mit acht ansteuerbaren Nadeln verfügt.

Wenn Sie keine Änderungen am Programm vornehmen möchten, sollten Sie den MSE-Lader (Listing 1) abtippen. Wollen Sie aber Änderungen und Verbesserungen daran vornehmen, verwenden Sie besser das Quellisting (Listing 2) und einen Assembler. Aber das werden Sie dann sowieso machen. Da in jeder Zeile nur ein einziger Befehl vorkommt, sollte das Quellisting ohne weiteres auf jeden Assembler, beispielsweise dem Hypra-Ass, übertragen werden können. Es müssen ja prinzipiell nur die Assembler-Anweisungen in den Zeilen 1000, 1010, 1100, 1110, 4070 und 4080 geändert werden. Die Bedeutung dieser Zeilen:

1000 Programmdatei für Maschinen-Code öffnen
1010 Starten des Assemblers
1100 Maschinen-Code in offene Datei schreiben
1110 Startadresse des Programms

4070 Ende des Quell-Codes
4080 Initialisierung der Hardcopy-Routine (nur sinnvoll, wenn der Maschinen-Code in den Speicher geschrieben wird und nicht auf Diskette gespeichert wird).

Hardcopy des Text-Bildschirms

Aber nun Schluß mit den allgemeinen Hinweisen. Im folgenden wird das Source-Listing anhand der Zeilennummern erklärt. Ab Zeile 1150 im Sourcelisting (Quellisting), werden den benötigten Betriebssystem-Routinen und Speicherzellen Labels (Namen) zugeordnet. Was die einzelnen Routinen leisten, entnehmen Sie bitte den Kommentaren im Quellisting.

Eine Hardcopy-Funktion für Textbildschirme wird erst dann so richtig interessant, wenn sie durch einfachen Tastendruck gestartet werden kann. Unabhängig davon, was der Computer gerade macht. Denn nur so kann man sich Beispiele von Bildschirmmasken etc. drucken lassen. Aus diesem Grund wurde dieser Hardcopy-Routine eine Tastenabfrage vorgeschaltet. So wird erreicht, daß durch kurzes Drücken der F1-Taste das Hardcopy-Programm startet. Wie man eine solche Tastenabfrage realisieren kann, sehen Sie ab Zeile 1460 im Quellisting. Dort wird die Abfrage-Routine in den Systeminterrupt des C 64 eingebunden. Danach wird die Taste pro Sekunde etwa 60mal abgefragt. Ist sie nicht gedrückt, wird der normale Systeminterrupt fortgesetzt. Zum Einbinden einer Routine in den Systeminterrupt (IRQ) muß der Interruptvektor in den Speicherzellen $314 und $315 auf die Routine »umgeleitet« werden. Im Normalfall zeigt der IRQ-Vektor auf den Beginn der Systeminterrupt-Routine bei $EA31. In diesem Fall wird er so verändert, daß er auf die Tastenabfrage, also auf $C00D zeigt. Genaueres zu diesem Thema können Sie beispielsweise im 64'er Sonderheft 4/85 nachlesen.

Die Veränderung des IRQ-Vektors wird in den Zeilen 1470 bis 1550 vorgenommen. Bevor aber der Vektor verändert werden kann, ist ein SEI(set-interrupt-disable-flag)-Befehl nötig. Dieser Befehl setzt das IRQ (interrupt-request)-Flag. Das bewirkt, daß der Prozessor keinen Interrupt mehr annimmt. Es ist ja eigentlich einleuchtend, daß kein Interrupt stattfinden darf, wenn man gerade die IRQ-Sprungadresse verändert. Das wäre so, als ob Sie auf einer Kreuzung ein Umleitungsschild aufstellen wollen, wenn dort gerade ein Lastzug fährt. Sie müssen vorher schon den Verkehr anhalten. Die Umstellung des IRQ-Vektors geschieht in diesem Programm durch SYS 49152. Dadurch wird die ganze Hardcopy-Routine initialisiert. Von nun an springt der Prozessor bei jedem Interrupt in die Tastenabfrage ab Zeile 1570. Dort wird der Inhalt der Speicherzelle $C5 geladen und mit »4« verglichen. »4« ist der Tastencode der F1-Taste. Fällt der Vergleich negativ aus, erfolgt sofort ein Sprung zur Adresse $EA31, der Systeminterrupt-Routine des C 64 (Zeile 1640). Fällt der Vergleich dagegen positiv aus (F1 gedrückt), verzweigt die Abfrage-Routine ins eigentliche Hardcopy-Programm ab Zeile 1700.

Start auf Tastendruck

Am Anfang der Hardcopy-Routine wird als erstes die gesamte Zeropage (Speicherzellen 0-255) zwischengespeichert. Dieses Verfahren ist zwar alles andere als elegant, erspart aber sehr viel Denkarbeit. Der Zeitbedarf dafür ist so gering, daß er bei einer Hardcopy-Routine vernachlässigt werden kann.

Nach dem »Retten« der Zeropage wird der Druckerkanal eröffnet und der Drucker normiert (Zeile 1670 bis 2000). Der Druckerkanal hat die Filenummer 126 und die Sekundäradresse 1. Die Geräteadresse ist 4. Die File-Nummer wurde so hoch gewählt, damit es keine Schwierigkeiten mit eventuell schon vorhandenen offenen Dateien gibt. Wer verwendet in Programmen schon die File-Nummer 126? Achtung: Eine File-Nummer größer als 127 sendet ein Linefeed (Zeilenvorschub) »CHR$(10)« nach jedem Carriage Return.

Ist der Druckerkanal offen, wird der Zeilenabstand des Druckers so eingestellt, daß alle Zeilen dicht aneinander liegen. Beim Epson RX/FX geschieht das mit der Steuersequenz CHR$(27)"3"CHR$(24). Mit der PRINT-Routine wird ein Zeichen an das, durch die CHKOUT-Routine festgelegte, Gerät geschickt. In diesem Fall eben an den Drucker.

Ab Zeile 2040 wird die Anfangsadresse des Bildschirms geholt und in VIRAM und VIRAM+1 übergeben. Das Programm muß ja wissen, wo die zu druckenden Daten stehen. Nach dieser Vorarbeit steht der eigentlichen Hardcopy-Routine nichts mehr im Weg.

In Zeile 2140 wird das x-Register mit der Zeilenzahl (25) des Bildschirms geladen. Danach wird die STOP-Taste abgefragt, um die Hardcopy vorzeitig beenden zu können. Die Abfrage findet übrigens zu Beginn jeder neu zu druckenden Bildschirmzeile statt. Ist die STOP-Taste gedrückt, wird die Hardcopy-Routine beendet und die normale Interrupt-Routine abgearbeitet. Wie die Hardcopy-Routine beendet wird, soll später erklärt werden. Vorläufig wird erstmal gedruckt!

Damit der richtige Zeichensatz aufs Papier kommt, wird ab Zeile 2390 bei jeder neuen Zeile ein Prüfprogramm (ab Zeile 3820, CHRTEST) aufgerufen, das den momentanen Schriftmodus überprüft. Groß-/Klein oder Groß-/Grafik-Zeichensatz. Die Startadresse des Zeichensatzes im Character-ROM wird dann in ZROM, ZROM+1 abgelegt. Der Groß/Klein-Zeichensatz hat die Startadresse $D800, der Groß/Grafik-Zeichensatz steht ab Adresse $D000. Welcher Zeichensatz gewählt ist, steht in Speicherzelle $D018 des Videochips. Ergibt der Inhalt von $D018 AND 2 das Ergebnis 2, ist der Groß/Grafik-Modus eingeschaltet. In Zeile 3890 finden Sie den BIT-Befehl $2C. Dieser Befehl vergleicht normalerweise den Akkuinhalt mit der angegebenen 2-Byte-Adresse und verändert entsprechend das Z-Flag. Diese Funktion ist aber hier nur Mittel zum Zweck. In diesem Fall soll der Prozessor nur die auf $2C folgenden 2 Bytes ignorieren. Genauer: Wird der Akku mit dem High-Byte der Startadresse des Zeichensatzes 1 geladen (Zeile 3870), wird der 2-Byte-Ladebefehl LDA # >CHRGEN2 (Startadresse Zeichensatz #2) einfach ignoriert. Man erspart sich dadurch eine zusätzliche Abfrage und einen Branch-Befehl.

Nachdem der aktuelle Zeichensatz bekannt ist, wird der Drucker in den Bitmustermodus (Grafikmodus) geschaltet. Dazu werden aus der Tabelle GRAFIK sieben Steuercodes geholt und zeichenweise an den Drucker geschickt. Die Steuersequenz ist 24, 13, 27, "*", 4, 64, 1. »1« und »64« ist das High- beziehungsweise das Low-Byte von 320, der Anzahl der Grafikpunkte des C 64 in einer Zeile. Der Code »13« bewirkt nur einen Wagenrücklauf (Carriage Return), damit der Druckkopf am Zeilenanfang steht. CHR$(24) löst einen Drucker-Reset aus.

In zwei Schleifen wird nun der ganze Bildschirm »abgetastet« und die einzelnen Zeichen am Bildschirm in Bitmusterdaten übersetzt. Den Anfang der übergeordneten »Zeilenschleife« lernten Sie bereits kennen: LDX #25, STOP-Taste abfragen, Druckerzeile initialisieren (in Grafikmodus schalten). Die untergeordnete »Spaltenschleife« (Zeilen 2530 bis 2650) holt nacheinander den Bildschirmcode der Zeichen einer Zeile. Der Zeichencode wird in der Speicherzelle ZEICHEN abgelegt. Als Zählvariable für diese untergeordnete Schleife, die von 0 bis 39 zählt, wird das y-Register hergenommen. Der Abschnitt von Zeile 2250 bis 2330 überprüft, ob in einer Zeile überhaupt etwas steht. Wenn nicht, wird ein Zeilenvorschub ausgelöst und die nächste Zeile »abgetastet«.

Ist ein Zeichen-Code ermittelt, wird mit JRS AUSWERT in

Zeile 2620 ein Unterprogramm aufgerufen, das aus dem Zeichensatz-ROM die Punktmuster der Zeichen liest. Dazu werden nacheinander die acht senkrechten Punktspalten eines Zeichens aus den Daten des Zeichensatz-ROMs zusammengesetzt. Jede fertige Punktspalte wird an den Drucker geschickt. Aus acht dieser Spalten entsteht dann ein Zeichen auf dem Papier, aus 320 solcher Spalten eine ganze Druckzeile. Das Lesen des Zeichensatz-ROMs und der Zusammenbau der Punktspalten ist für den Anfänger in Maschinensprache nicht ganz einfach zu verstehen und soll deshalb kurz erklärt werden.

Das Unterprogramm AUSWERT beginnt ab Zeile 3060 mit drei PHA (push akku)- und zwei Transferbefehlen, die den Inhalt des Akkus und des x-,y-Registers auf den Prozessorstack legen. Das ist nötig, da im Unterprogramm der Akku und das x- und y-Register verändert werden. Danach wird in den Zeilen 3160 bis 3290 die Startadresse des momentanen Zeichens im Zeichensatz-ROM berechnet. Die Startadresse berechnet Anfangsadresse + Offset (Speicherstelle im Zeichensatz-ROM). Offset = Bildschirm-Code x 8.

Wie schon erwähnt, wird ab Zeile 2530 der »Spaltenschleife« der Bildschirmcode eines jeden Zeichens am Bildschirm geholt (LDA (VRAM,Y)) und in ZEICHEN gespeichert. Der Inhalt von ZEICHEN wird in Zeile 3170 in die Speicherzelle ADRESSE kopiert. Dann wird durch drei ASL-Befehle der Inhalt Speicherzelle ADRESSE und ADRESSE+1 (Grundwert immer »0«) mit 8 multipliziert. Ein ASL (arithmical shift left) verdoppelt den Inhalt einer Speicherzelle. Tritt dabei ein Übertrag auf (Ergebnis > 255) wird das Carry-Flag gesetzt. Das Carry-Flag wird in der Speicherzelle ADRESSE+1 berücksichtigt. Ein ROL-Befehl schiebt das Carry-Bit »von rechts in die Speicherzelle ADR+1«. Fand kein Übertrag statt, schiebt der ROL-Befehl einfach eine »0« nach. Der ROL-Befehl hat die gleiche Wirkung, wie eine Verdopplung eines Speicherzelleninhalts mit ASL, nur wird anschließend noch das Carry-Bit addiert. Nach der Multiplikation mit 8 steht in ADRESSE und ADRESSE+1 in Low-/High-Bytedarstellung der Offset. Um die absolute Adresse zu erhalten, wird ab Zeile 3110 der Offset zur Anfangsadresse des Zeichensatz-ROMs addiert. Es steht dann die absolute Zeichenadresse in ADRESSE (Low-Byte) und ADRESSE+1 (High-Byte).

Der Clear-Carry (CLC)-Befehl ist grundsätzlich vor jeder Addition notwendig, um ein eventuell gesetztes Carry-Flag zu löschen. Nur so kann ein Übertrag sicher festgestellt werden.

Ist die Anfangsadresse eines Bildschirmzeichens im Zeichensatz-ROM bekannt, fängt aber die Arbeit erst richtig an.

Ein Bildschirmzeichen setzt sich aus einer 8x8-Punktmatrix zusammen. Die Punktmatrix wird beim C 64 aus acht 8-Bit-Zahlen zusammengesetzt:

Adresse		Adresse	
$D000	....	$D005	....
$D001	....	$D006	....
$D002	....	$D007	....
$D003	....	$D008	....
$D004	....	$D009	....

Mit einer 8-Bit-Zahl pro Punktzeile. Dabei bestimmt die erste Zahl im ROM das Punktmuster der obersten Punktzeile eines Zeichens. Stellen Sie sich eine 8-Bit-Zahl einmal in Binärform vor. Eine »1« ist dann ein gesetzter Punkt und eine »0« kein Punkt. So könnte die Zahl 0000 1000 (dez. 8) einen i-Punkt wiedergeben. Wie Sie sehen, sind die Bildschirmzeichen aus Punktzeilen aufgebaut; im Gegensatz zu den Druckerzeichen, die aus Punktspalten zusammengesetzt werden. Die beiden Formate bringen ein Problem mit sich, denn die Bitmusterdaten des Bildschirms müssen in die des Druckers übersetzt werden. Diese Konvertierung wird in den Zeilen 3330 bis 3660 vorgenommen.

Doch bevor man die Bitmusterdaten konvertiert, muß erstmal eine Punktzeile aus dem Zeichensatz-ROM gelesen werden. Wie Sie vielleicht wissen, liegt das Zeichensatz-ROM im $D000-Bereich, dem kompliziertesten Speicherteil des C 64. Im Bereich von $D000 bis $DFFF arbeiten nämlich noch alle I/O-Bausteine. Man spricht in diesem Zusammenhang von Speicherebenen, die sich einen Speicherbereich teilen. Damit der Zeichensatz gelesen werden kann, muß also noch die richtige Speicherebene selektiert werden. Doch dazu später mehr. Zuerst soll die Konvertierung der Bitmusterdaten geklärt werden.

Zur Konvertierung der Zeilenwerte in Spaltenwerte liest man nacheinander acht mal die Zeilenwerte des Zeichensatz-ROMs und vergleicht die Werte mit einer »Bit-Maske«. Die acht Bit-Masken sind:

Maske bin.	Wertigkeit
1000 0000	128
0100 0000	64
0010 0000	32
0001 0000	16
0000 1000	8
0000 0100	4
0000 0010	2
0000 0001	1

Beim ersten Durchgang hat die Bitmaske die Wertigkeit 128 (Zeile 3360). Mit dieser Maske werden nun nacheinander alle acht Punktzeilen eines Bildschirmzeichens AND-verknüpft. Ist beispielsweise in der Zeilenzahl das achte Bit gesetzt (Zahl >127), ist das Ergebnis der Verknüpfung 128, also größer 0. Liefert die Verknüpfung »0«, wird der nächste der acht Zeilenwerte mit der Maske verglichen. Ist das Ergebnis einer Verknüpfung größer als 0, muß an dieser Stelle eine Druckernadel anschlagen.

Bei einem 8-Nadeldrucker haben die Nadeln die binären Werte 1,2,4,8,16,32,64,128. Die unterste Nadel hat die Wertigkeit 1, die oberste die Wertigkeit 128. Ausnahmen bestätigen aber auch hier die Regel (Seikosha 550A: Die untere Nadel hat die Wertigkeit 128). Damit der Drucker die richtige Nadel anschlägt, muß die Wertigkeit der Nadel, oder die Summe der Wertigkeiten, dem Drucker geschickt werden. Dazu ordnet man den acht Punktzeilen eines Bildschirmzeichens die binären Wertigkeiten 1 bis 128 zu; entsprechend der Nadelanordnung des Druckers. Bei jedem positiv ausgefallenen Maskenvergleich, addiert man die entsprechenden Nadelwertigkeiten. Wurden alle acht Zeichenzeilen auf diese Weise »abgetastet«, wird die Summe an den Drucker geschickt, der daraufhin eine senkrechte Punktreihe druckt.

Um die nächste Punktspalte drucken zu können, dividiert man die Maske mit »2«. Am einfachsten mit einem LSR (logical shift right)-Befehl. Dieser Befehl verschiebt die »1« in der »Binärmaske« um eine Stelle nach links, was eben einer Division mit 2 gleichkommt. Nun vergleicht man wieder die acht Zeilenwerte mit der neuen Maske und addiert die Nadelwertigkeiten. Nach acht Vergleichen ist der Druckercode für die zweite Druckspalte addiert und kann zu Papier gebracht werden. Dieses Spielchen wiederholt man, bis alle acht Punktspalten eines Zeichens gedruckt sind. Für ein einziges Zeichen sind insgesamt 64 Vergleiche nötig.

Als Basic-Programm wäre die Konvertierung der Bitmusterdaten des C 64 in die eines Druckers viel zu langsam. Es sind schließlich 64x40x25=64000 Konvertierungen pro Bildschirm nötig. Ebenso ist die »Bit-Schieberei« mit LSR, OR und AND in Basic ein Problem für sich.

Die Speicherebenenumschaltung auf das Zeichensatz-ROM findet in Zeile 3450 statt, nachdem zuvor das Interrupt-



Flag gesetzt wurde. Denn solange Speicherzelle 1 den Wert $33 enthält, dürfen keine I/O-Operationen des Prozessors, wie Tastenabfrage und Cursor-Blinken, erfolgen. Der Prozessor würde unweigerlich abstürzen, wenn er auf das Zeichensatz-ROM zugreift, anstelle auf einer der beiden CIAs oder dem VIC-Chip.

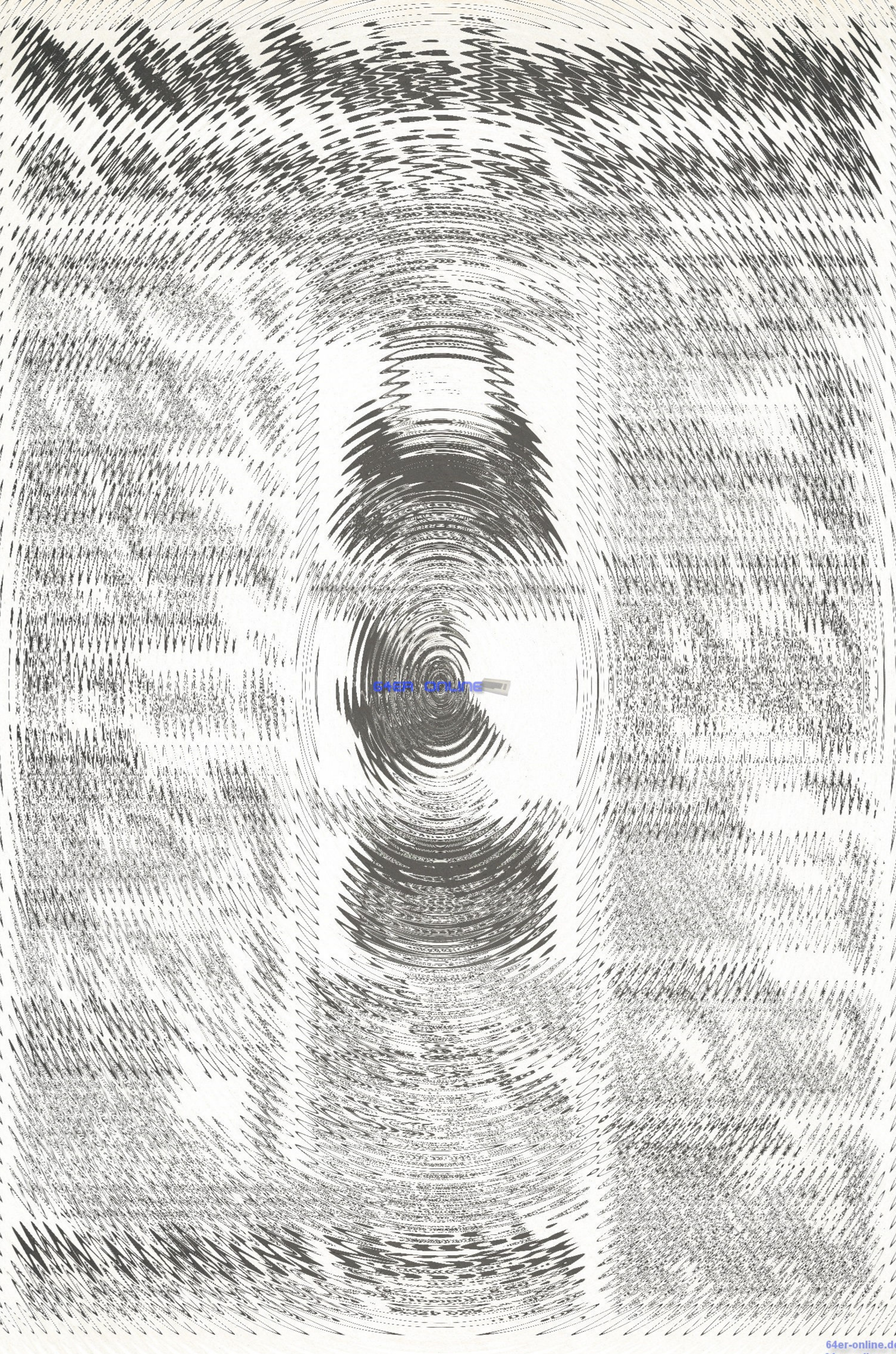
In Zeile 3480 wird das Zeichensatz-ROM ausgelesen und in Zeile 5335 mit der aktuellen Maske verglichen. Danach wird

Fortsetzung auf Seite 177

```
1000 open2,8,2,"hardcopy.obj,p,w"
1010 sys9*4096
1020 ;*********************************
1030 ;*      lowscreenhardcopy         *
1040 ;*            incl.               *
1050 ;*   grafik- und reverszeichen    *
1060 ;*     epson/wiesemann & komp.    *
1070 ;*         version 1.2            *
1080 ;*    harald meyer 21.05.1985     *
1090 ;*********************************
1100 .opt o2  ; code auf disk
1110 *= $c000 ; startadresse
1120 ; routinen des betriebssytems
1130 ;**********************************
1140 ;
1150 open    = $ffc0 ; file oeffnen
1160 setnam  = $ffbd ; filenamen setzen
1170 setfls  = $ffba ; fileparameter
1180 print   = $ffd2 ; zeichen ausgeben
1190 clrch   = $ffcc ; bildsch.-ausgabe
1200 chkout  = $ffc9 ; ausgabegeraet
1210 close   = $ffc3 ; file schliessen
1220 stop    = $ffe1 ; stopvektor
1230 irqend  = $ea31 ; kernal-irq-rout.
1240 lovideo = $d018 ; videoram lo
1250 chrgen1 = $d000 ; 1.zeichensatz
1260 chrgen2 = $d800 ; 2.zeichensatz
1270 ;
1280 ; benoetigte speicherzellen
1290 ; *********************************
1300 ;
1310 cursor  = 204   ; cursor aus/an
1320 irqvek=$0314;irq-vektor
1330 taste   = $c5   ; letzte taste
1340 f1      = 4     ; f1-matrixnummer
1350 cr      = 13    ; carriage return
1360 esc     = 27    ; escape
1370 adresse = $f8   ; zeichenadresse
1380 zeichen = $d6   ; zeichencode
1390 viram   = $15   ; zeichenadresse
1400 zrom    = $f9   ; zeichengenerator
1410 maske   = $9d   ; bit-abfrage
1420 ;
1430 ; interrupt initialisieren
1440 ; *********************************
1450 ;
1460 sei             ; irq verhindern
1470 lda #<start     ; irq vektor auf
1480 ldy #>start     ; dieses programm
1490 sta irqvek
1500 sty irqvek+1
1510 cli             ; irq freigeben
1520 rts             ; init. ende
1530 ;
1540 ;
1550 ; f1 gedrueckt, dann start
1560 ; *********************************
1570 ;
1580 start lda taste ; tastenabfrage
1590 cmp #f1         ; taste = f1 "?"
1600 bne l1          ; nein, dann ende
1610 lda #00         ; tastendruck
1620 sta taste       ; loeschen
1630 jsr hdcopy      ; programmbeginn
1640 l1  jmp irqend  ; zum kernal-irq
1650 ;
1660 ;
1670 ; zeropage retten
1680 ; *********************************
1690 ;
1700 hdcopy ldx #$ff ; byte 255-0
1710 l2  lda 0,x     ; laden
1720 sta memory,x    ; und speichern
1730 dex             ; naechstes byte
1740 bne l2          ; x=0, dann ende
1750 lda #01
1760 sta cursor      ;ausschalten
1770 ;
1780 ;
1790 ; druckerfile oeffen und
1800 ; zeilenabstand initialisieren
1810 ; *********************************
1820 lda #126      ; filenummer
1830 ldx #4        ; geraeteadresse
1840 ldy #1        ; sekundaeradresse
1850 jsr setfls    ; parameter setzen
1860 lda #0        ; kein filenamen
1870 jsr setnam    ; namen setzen
1880 jsr open      ; file oeffnen
1890 ldx #126      ; alle
1900 jsr chkout    ; ausgaben auf #126
1910 ;
1920 ; drucker auf einzeiligen abstand
1930 lda #esc      ; esc-sequenz-beginn
1940 jsr print     ; chr$(27) an drucker
1950 lda #"3"      ; "3"+chr$(24)
1960 jsr print
1970 lda #24
1980 jsr print     ; an drucker
1990 ;
2000 ; zeiger auf videoram
2010 ; *********************************
2020 ;
2030 videoram lda #$00
2040 ldy $288      ; videoramadressen
2050 sta viram     ; uebergeben
2060 sty viram+1
2070 ;
2080 ;
2090 ; zeichen vom screen holen,
2100 ; charaktergenerator lesen und
2110 ; zeichenmatrix zusammensetzen
2120 ; *********************************
2130 ;
2140 ldx #25       ; zeilenanzahl
2150 ;
2160 ; ausgabeschleife
2170 ; *********************************
2180 ;
2190 ausg jsr stop; stoptaste abfragen
2200 beq hdende    ; gedrueckt,dann ende
2210 ;
2220 ; ist zeile leer "?"
2230 ; *********************************
2240 ;
2250 ldy #39
2260 l9 lda (viram),y
2270 cmp #32
2280 bne l10
2290 dey
2300 bpl l9
2310 lda #cr
2320 jsr print
2330 jmp l11
2340 ;
2350 ; chrgen-adresse holen
2360 ; gross/klein oder gross/grafic
2370 ; *********************************
2380 ;
2390 l10 jsr chrtest ;schriftmodus-test
2400 ;
2410 ; druckzeile vorbereiten
2420 ; *********************************
2430 ;
2440 ldy #7               ; 6 codes
2450 l3  lda grafik,y     ; esc-sequenz
2460 jsr print            ; senden
2470 dey                  ; naechster code
2480 bne l3               ; fertig "?"
2490 ;
2500 ; neue zeile beginnen
2510 ; *********************************
2520 ;
2530 ldy #0  ; zeile von neuem
2540 ;
2550 ; zeile abarbeiten, 2. schleife
2560 ; zeichen holen und im up bearbei.
2570 ; *********************************
2580 ; zeichen v. bildsch. holen
2590 l5  lda #0   ; zeichen von
2600 lda (viram),y; bildschirm holen
2610 sta zeichen  ; merken
2620 jsr auswert  ; bearbeiten
2630 iny          ; spalte erhoehen
2640 cpy #40      ; zeilenende "?"
2650 bne l5       ; bildsch.-ende "?"
2660 ;
2670 ; neue zeile vorbereiten
2680 ; *********************************
2690 ;
2700 l11 lda #40; 40 spalten
2710 clc     ; zeiger auf naechste zeile
2720 adc viram    ; neue zeile setzen
2730 sta viram
2740 bcc l12
2750 inc viram + 1
2760 l12 dex          ; zeilen erniedrigen
2770 bne ausg         ; screen zu ende "?"
2780 hdende lda #13; zum ende cr an
2790 jsr print        ; drucker senden
2800 ;
2810 ; fertig, dann file schliessen
2820 ; *********************************
2830 ;
2840 lda #126 ; filenummer
2850 jsr close; druckerdatei schliessen
2860 jsr clrch; cmd auf screen
2870 ;
2880 ;
2890 ; zeropage wiederherstellen
2900 ; *********************************
2910 ;
2920 ldx #$ff          ; zeropage
2930 l4  lda memory,x; wieder
2940 sta $00,x         ; herstellen
2950 dex
2960 bne l4
2970 ;
2980 rts               ; hardcopy ende
2990 ;
3000 ;
3010 ;
3020 ;
3030 ; charaktergenerator lesen
3040 ; *********************************
3050 ;
3060 auswert pha;  register retten
3070 txa
3080 pha
3090 tya
3100 pha
3110 ;
3120 ; adresse im charakterram berech.
3130 ; = zeichencode * 8
3140 ; *********************************
3150 ;
3160 lda zeichen; zeichencode laden
3170 sta adresse; adresse im zeichen-
3180 lda #0     ; rom feststellen
3190 sta adresse+1
3200 asl adresse; zeichencode
3210 rol adresse+1
3220 asl adresse
3230 rol adresse+1
3240 asl adresse
3250 rol adresse+1 ; mal 8
3260 lda adresse+1
3270 clc               ; und
3280 adc zrom+1     ; romadresse
3290 sta adresse+1 ; addieren
3300 ;
3310 ; charaktergen. lesen und zeichen-
3320 ; matrix fuer drucker aufbereiten
3330 ; *********************************
3340 ;
3350 ldx $01          ; speicherselekt
3360 lda #%10000000 ; bit-maske
3370 sta maske        ; speichern
3380 l14 lda #$00     ; code
3390 pha              ; loeschen
3400 ldy #7           ; 8 bytes abfragen
3410 ;
3420 ; chr-ram einschalten & byte lesen
3430 ; *********************************
3440 ;
3450 l15 sei     ; irq sperren
3460 lda #01:and #251          ; zeichen-rom
3470 sta $01     ; selektieren
3480 lda (adresse),y ; code holen
3490 and maske         ; maskenvergleich
3500 ;
3510 ; chr-rom wieder einschalten
3520 ; *********************************
3530 ;
3540 stx $01        ; norm speicherkonf.
3550 cli            ; wieder herstellen
3560 beq l6
3570 pla            ; code laden und
3580 clc            ; code laden und
3590 adc nwert,y ; bitwert zu code add.
3600 pha            ; und merken
3610 l6  dey        ; naechstes byte
3620 bpl l15        ; 8 bits gelesen "?"
3630 pla            ; ja, dann
3640 jsr print      ; an drucker
3650 lsr maske      ; maske erhoehen
3660 bcc l14        ; naechste punktzeile
3670 ;
3680 ; register wieder holen
3690 ; *********************************
3700 ;
3710 pla ; register wieder holen
3720 tay
3730 pla
3740 tax
3750 pla
3760 rts
3770 ;
3780 ; anfangsadresse des charakterrams
3790 ; feststellen und merken
3800 ; *********************************
3810 ;
3820 chrtest lda #$00
3830 sta zrom      ; zeichensatzadr. lo
3840 lda lovideo   ; zeichensatz-page
3850 and #%00000010
3860 bne l16
3870 lda #>chrgen1          ;$d000
3880 sta zrom + 1
3890 .byt $2c
3900 l16 lda #>chrgen2      ;$d800
3910 sta zrom + 1
3920 rts                ; zum hauptprogramm
3930 ;
3940 ;
3950 ;
3960 ;tabelle 320 bitmusterdaten
3970 grafik .byt 0,$01,$40
3980 ;
3990 ;epson auf einzelnadelansteuerung
4000 .byt 4,"*",esc,cr,24
4010 ;
4020 ;tabelle fuer bit-wertigkeiten
4030 nwert .byt 128,64,32,16,8,4,2,1
4040 ;
4050 ;zeichensatz startadressen
4060 memory .byt 0
4070 .end
4080 sys49152

ready.
```

Quelltext der Hardcopy-Routine



Leben und sterben lassen

»Life« ist wohl eine der faszinierendsten Simulationen von biologischen Vorgängen auf Computern. Mit »Life« lassen sich evolutionäre Abläufe spielerisch verstehen lernen.

Obwohl den Lesern der Inhalt des »Spiels« geläufig sein mag, hier die »Regeln«:
Es existiert ein zweidimensionales Feld und an jedem Punkt ist eine lebende oder eine tote Zelle. Jede Zelle hat acht Nachbarn.
An einem Punkt wird eine Zelle dann und nur dann lebendig, falls sie genau drei lebende Nachbarn hat.

Fast wie im richtigen Leben

Eine Zelle stirbt, wenn sie weniger als zwei (Vereinsamung) oder mehr als drei (Überbevölkerung) lebende Nachbarn hat.
So ändert sich das Feld von Generation zu Generation, mit einer meist faszinierenden Entwicklung. Das Spiel, erfunden von dem Mathematiker John Horton Conway (Cambridge), stellt sich als höchst komplex und reizvoll heraus.
Um es gleich zu sagen: Life ist kein Spiel, das man selber spielt. Life spielt sich selbst. Man kann die »Anfangswelt« festlegen und dann beobachten, wie sie sich entwickelt. Das Spiel wird sicher dem einen oder anderen Computer-Fan bekannt sein, da es ein ideales Problem für den Computer ist: Sehr einfache Regeln, aber viel Rechnerei.
In dem Beispiel (Rotor, siehe Bild 1) sind die Verhältnisse noch sehr einfach, weil die kleine Life-Welt nur zwischen zwei Zuständen hin und her pendelt. Dieser Dreier ist eine der Strukturen, die stabil sind. Meistens dauert es viele Generationen, bis eine Population ausstirbt oder in stabile Strukturen übergeht. Am besten, man experimentiert einfach etwas. Bild 2 zeigt noch einige interessante Strukturen.

Zum Programm:

Wie würde man das Problem normalerweise angehen? Nun, man legt sich ein zweidimensionales Feld an und speichert für jede lebende Zelle eine Eins, für jedes unbesetzte Kästchen eine Null. Dann geht man alle Felder durch, prüft die acht Nach-



Bild 1. Ein einfaches Beispiel erläutert die Funktionsweise von »Life«

stabile Strukturen (Zyklus 1 Generation)

Gleiter (Zyklus 4 Generationen dabei sich fortbewegend)

Gleiterkanone (Zyklus ca. 30 Generationen)

Kaleidoskop (3er Zyklus nach ca. 20 Generationen)

Evolution (Endstadium nach ca. 100 Generationen)

Bild 2. Anhand einfacher Strukturen wird die Faszination von »Life« deutlich

Routinen des Maschinenprogramms

$C000-C023	16-Bit-Multiplikation (Listing 2)
$C024-C067	Rechnung: 6000+3 (X+40 Y)→SP(16B)
$C068-C079	Rechnung: 4 S(16B)→SP1(16B)
$C080-C097	Rechnung: 4 M(16B)→SP1(16B)
$C098-C0A5	Inkrement X — Verändert SP(16B) entsprechend,
$C0A6-C0B3	Inkrement Y — so daß es dem Ergebnis der
$C0B4-C0C1	Dekrement X — Routine $C024 entspricht.
$C0C2-C0CF	Dekrement Y
$C0D0-C0DC	erhöht M, falls Zelle bei X,Y lebt
$C0DD-C104	relevanter Bereich, speichert Koordinaten
$C105-C15C	relevanter Bereich, geht neun Umgebungszellen durch, ruft $C0DD auf
$C15D-C2A4	nächste Generation, läuft zweimal durch alle Koordinaten des relevanten Bereichs
$C2A5-C332	Plot: setzt, löscht und prüft Punkte im 80 x 50-Bild
$C333-C3A6	Initialisierungsroutine
$C3A7-C3C4	Aufruf relevanter Bereich, setzt A%(X,Y)=1, trägt Koordinaten der Umgebungsfelder ein (SYS C3A7,X,Y)
$C3C5-C3F6	Aufruf Plot, SYS C3C5,X,Y,Z, löscht, setzt (Z30 = oder 1) oder kontrolliert (Z=2) Punkte
$C3F7-C3FC	Aufruf nächste Generation, ruft $C15D auf, setzt zwecks Beschleunigung Interrupt-Flag
$C3FD-C403	Aufruf maximale Geschwindigkeit, setzt I-Flag, ruft in Endlosschleife $C15D

Variablen des Maschinenprogramms

$C404-C413	Tabelle Binärzahlen→ASCII-Code (der Grafikzeichen für Plot)
$C414	Z (siehe $C3C5), enthält wegen logischen Verknüpfungen 00 für setzen und $OF für löschen
$C415	L gibt an, welches Bit des Grafikzeichens angesprochen wird
$C416	QQ liefert Ergebnis des Tests
$C417/8	X/Y
$C419/B	Zwischenspeicher1/2
$C41A	E gibt an, welcher A%() Speicherbedarf angesprochen wird
$C41C/D	M(16B) — Zahl der Koordinaten be-
$C41E/F	S(16B) — ziehungsweise Schleifenindex
$C420	M Zahl der lebenden Zellen in jeweiliger Umgebung
$C421	J gibt an, welcher X%,Y% Speicherbetrieb angesprochen wird
$C422-C521	Tabelle ASCII-Code→Binärzahlen (wird erzeugt)
$57-5B	für 16-Bit-Multiplikation (5B)x(59/5A)=(57/58)
$57/58	SP(16B) (lokal)
$59/5A	SP1(16B) (lokal)

S(16B) bedeutet Speicher S mit der Länge von 16 Bit. X und Y haben nichts mit den Registern zu tun, die Namen der Variablen beziehen sich auf das Basic-Programm (Listing 1)

Tabelle 1. Der Aufbau des Maschinenprogramms (Listing 2)

barfelder und speichert das Ergebnis, tot oder lebendig in der nächsten Generation, in einem zweiten Feld. Der Computer hat dann (bei 80 mal 50) 32 000 Felder zu überprüfen, und das bedeutet: warten ...

Aber Moment! Ist es denn nötig, auch die Bereiche zu bearbeiten, in denen es ohnehin kein Leben gibt?

Besser, man bearbeitet jeweils nur die relevanten (belebten) Bereiche: Es kann nur in dem Bereich etwas passieren, der direkt an lebende Zellen angrenzt. Man speichert die Koordinaten aller lebenden Zellen und ihrer Nachbarn, sorgt dafür, daß keine Felder doppelt gezählt werden und braucht dann jeweils nur noch die Liste der Koordinaten durchzugehen. Dafür benötigt man ein weiteres Variablen-Feld, in dem bereits vermerkte Zellen gekennzeichnet werden. Nach dieser Vorbereitung werden zur Berechnung der nächsten Generation die Koordinaten des relevanten Bereichs durchgegangen. Ein zweites Mal durchläuft man die Koordinaten, um den neuen relevanten Bereich festzustellen. Wenn also eine Struktur auf dem Bildschirm wandert, so wandert auch der relevante Bereich mit.

Bedienungshinweise:

Man hat eine 78 x 48-Auflösung (der äußerste Rand ist eine Todeszone«; hier ist grundsätzlich kein Leben möglich) zur Verfügung. Startet man das Programm, so befindet man sich in einem Editor.

Eine Zelle wird mit der SPACE-Taste lebendig.

Eine Zelle wird mit der DEL-Taste getötet.

Der Cursor schreitet nach SPACE und DEL in die Richtung fort, die die letzte gedrückte Cursortaste angab.

Mit »L« wird Life dann gestartet. In dem jetzigen Zustand gibt es folgende Befehle:

S Stop
F1-F8 Verzögerung
N Neue Eingabe
E Ergänzung, Veränderung
Q Quit, Programmende
P Power, maximale Geschwindigkeit (nur mit RUN-STOP/RESTORE zu lösen)

Erweiterungsmöglichkeiten:

Man könnte natürlich die Auflösung weiter erhöhen. Da böte sich dann die Bitmap-Auflösung an (HiRes-Grafik). Hier müßte man allerdings mit dem Speicherplatz sehr vorsichtig umgehen, da der Bildschirm 64000 Bildpunkte hat. Die zusätzlichen Berechnungen würden das Programm auf jeden Fall langsamer machen. Außerdem wird, wenn die relevanten Bereiche zu groß werden, ein Speichern derselben unmöglich (Speicherplatz) und der Generationswechsel dauert zu lange. Eine andere interessante Veränderung wäre mit dem Rand durchzuführen. Mit dem »Todesrand« wurde das Problem der Bildschirm-Bereichsüberschreitung zwar recht elegant gelöst, denn es sind durch das Prinzip der relevanten Bereiche keine Abfragen notwendig. Interessanter jedoch wäre eine offene Wand oder gar eine ganz in sich geschlossene Welt. Unten (rechts) heraustretende Formen würden dann oben (links) wieder auftauchen. Wegen der Abfragen würde jedoch auch dies die Geschwindigkeit verringern.

Hinweise zum Abtippen:

Als erstes geben Sie die Maschinenroutine (Listing 2) mit dem MSE ein (Speichern nicht vergessen). Zur Eingabe des Hauptprogramms (Listing 1) verwenden Sie bitte den Checksummer.

Interessierte Leser finden in der Tabelle 1 die Beschreibung des Maschinenprogramms.

Wer will, kann das Programm noch wesentlich erweitern: Zum Beispiel könnten Hungersnöte, Naturkatastrophen oder Seuchen in bestimmten Gebieten die Bevölkerung wesentlich dezimieren. Ich glaube nicht, daß man das Spiel »Life« auf dem C 64 noch wesentlich schneller programmieren kann, jedoch soll diese Behauptung ruhig dazu provozieren, das Gegenteil zu beweisen. (Jürgen Engbring/tr)

```
10 REM--------------------------                <165>
20 REM**    LIFE FUER C-64    **                 <182>
25 REM--------------------------                <180>
30 REM** JUERGEN ENGBRING     **                 <067>
40 REM** HENKELSHOF 5-7       **                 <166>
50 REM** 5630 REMSCHEID-11    **                 <195>
60 REM** TEL. 02191/65533     **                 <215>
70 REM--------------------------                <225>
80 M=49152:FOR I=0 TO 9:READ A                   <164>
85 IF A<>PEEK(M)THEN LOAD"MASCHLIFE",8,1         <171>
90 M=M+100:NEXT                                  <095>
95 DATA 169,88,87,206,2,152,0,192,45,0           <098>
100 PRINT"{CLR}"                                 <088>
110 SYS 49971: REM INITIALISIERUNG               <126>
120 PL=50117: REM SYSPL,X,Y,0/1/2                <105>
130 :N=50167: REM NAECHSTE GENERATION            <061>
140 :S=50173: REM MAXIMALE GESCHW.               <163>
150 :G=50087: REM SYSG,X,Y ZELLE  LEBT           <179>
155 BA=6*4096:REM BASIS ZELLSPEICHER             <247>
160 QQ=50198: REM TESTBYTE SYSPL,X,Y,2           <179>
170 REM+++++++++++++++++++++++++++++++++         <049>
990 REM===  80 X 40 ZELLEDITOR  ========         <080>
1000 DIM G%(255):X=39:Y=24:Q=TI:CH=29            <129>
1005 FOR I=0 TO 6:READ A:READ B:G%(A)=B:NE
     XT                                          <205>
1010 DATA 17,1,29,2,145,3,157,4,20,5,32,6,
     76,7                                        <170>
1020 GET A$:IF TI-Q>15 THEN Q=TI:W=1-W           <118>
1030 SYS PL,X,Y,W                                <141>
1040 IF A$=""THEN 1020                           <222>
1050 A=ASC(A$)                                   <184>
1060 ON G%(A)GOTO 1100,1200,1300,1400,1500
     ,1600,1700                                  <079>
1070 GOTO 1020                                   <058>
1100 SYS PL,X,Y,Z:Y=Y+1:CH=17                    <088>
1110 IF Y>49 THEN Y=0                            <150>
1120 GOTO 1450                                   <012>
1200 SYS PL,X,Y,Z:X=X+1:CH=29                    <013>
1210 IF X>79 THEN X=0                            <081>
1220 GOTO 1450                                   <112>
1300 SYS PL,X,Y,Z:Y=Y-1:CH=145                   <124>
1310 IF Y<0 THEN Y=49                            <244>
1320 GOTO 1450                                   <214>
1400 SYS PL,X,Y,Z:X=X-1:CH=157                   <209>
1410 IF X<0 THEN X=79                            <214>
1420 GOTO 1450                                   <058>
1450 SYS PL,X,Y,2:Z=PEEK(QQ)                     <197>
1460 GOTO 1020                                   <194>
1500 Z=0:A=CH:POKE BA+3*X+Y*240,0                <110>
1510 GOTO 1060                                   <116>
1600 Z=1:A=CH                                    <032>
1610 SYS G,X,Y:GOTO 1060                         <137>
1700 SYS PL,X,Y,Z                                <059>
2000 T=9999:REM MINIMALE VERZOEGERUNG            <014>
2005 POKE 50209,2:REM VOR ERSTEM SYSN            <081>
2010 GET A$:IF TI-Q>T THEN Q=TI:SYS N            <011>
2020 IF A$=""THEN 2010                           <178>
2030 IF A$=" "THEN SYS N:T=9999:GOTO 2010        <068>
2040 IF A$="S"THEN T=99999                       <059>
2050 IF A$="N"THEN RUN                           <044>
2055 IF A$="E"GOTO 2300                          <176>
2060 IF A$="Q"THEN:END                           <121>
2065 IF A$="P"THEN SYS S                         <038>
2067 IF A$=CHR$(133)GOTO 2200                    <086>
2070 A=ASC(A$)                                   <188>
2080 IF A<134 OR A>140 GOTO 2010                 <017>
2090 T=2*2↑(A-134)                               <159>
2100 GOTO 2010                                   <048>
2200 GET A$:IF A$=""THEN SYS N:GOTO 2200         <155>
2210 GOTO 2020                                   <190>
2300 IF PEEK(50202)=1 THEN SYS N                 <063>
2310 POKE 50209,0:GOTO 1020                      <141>

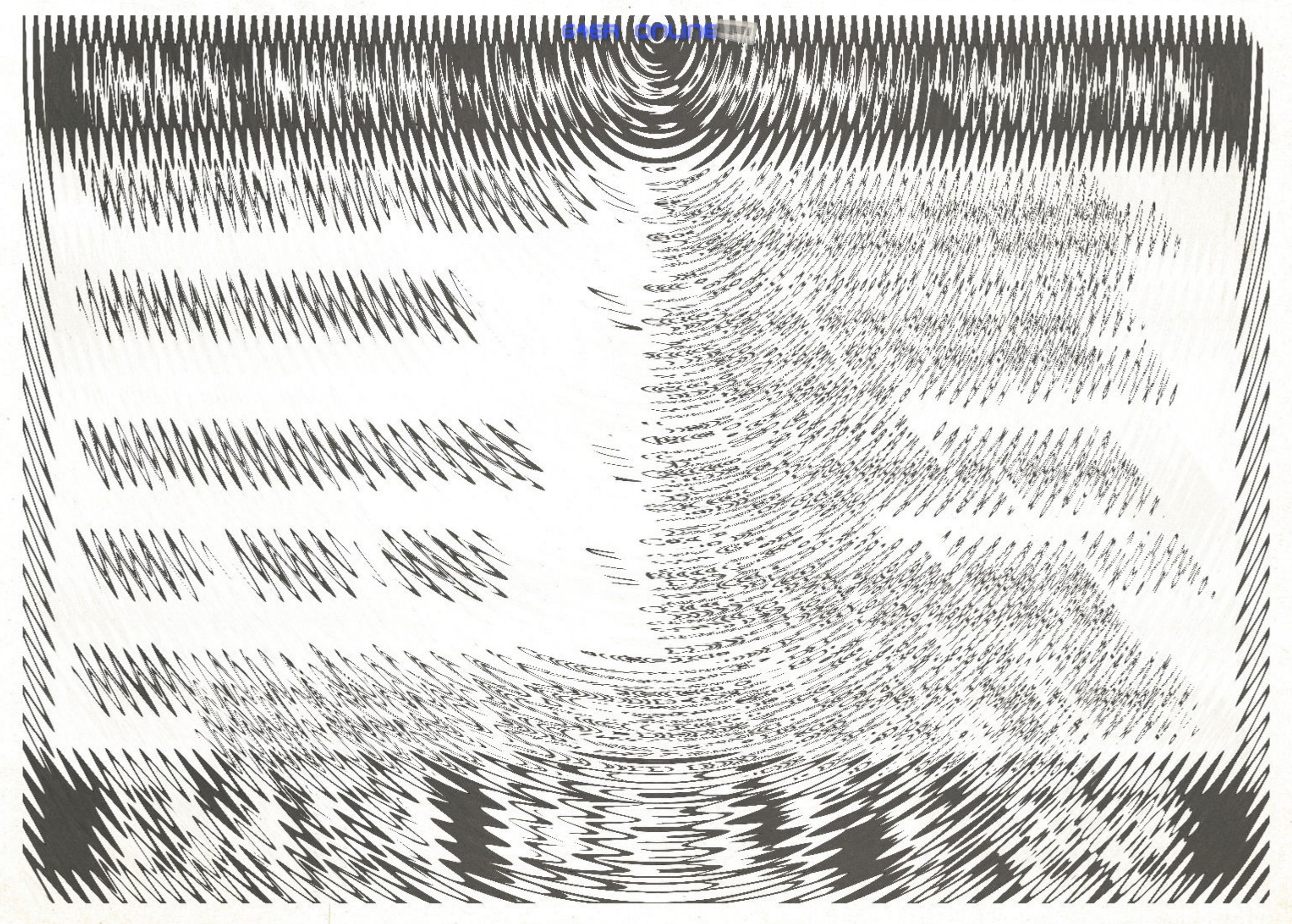
© 64'er
```

Listing 1. Der Basic-Teil von »Life« ist am besten mit dem Checksummer einzugeben

```
programm : maschlife              c000 c414
-------------------------------------------

c000 : a9 00 85 57 85 58 a0 08   a3
c008 : 18 26 5b 90 0d 18 a5 57   f3
c010 : 65 59 85 57 a5 58 65 5a   d5
c018 : 85 58 88 f0 06 26 57 26   45
c020 : 58 90 e6 60 a9 50 85 5b   70
c028 : a9 00 85 5a ad 18 c4 85   38
c030 : 59 20 00 c0 ad 17 c4 18   88
c038 : 65 57 85 57 a9 00 65 58   76
c040 : 85 58 a5 57 85 59 a5 58   b0
c048 : 85 5a 06 57 26 58 18 a5   38
c050 : 59 65 57 85 57 a5 5a 65   b9
c058 : 58 85 58 a9 00 65 57 85   52
c060 : 57 a9 60 65 58 85 58 60   24
c068 : ad 1e c4 0a 85 59 ad 1f   af
c070 : c4 2a 85 5a 06 59 26 5a   6e
c078 : 18 a9 40 65 5a 85 5a 60   1e
c080 : ad 1c c4 0a 85 59 ad 1d   c2
c088 : c4 2a 85 5a 06 59 26 5a   86
c090 : 18 a9 40 65 5a 85 5a 60   36
c098 : 18 a9 03 65 57 85 57 a9   45
c0a0 : 00 65 58 85 58 60 18 a9   56
c0a8 : f0 65 57 85 57 a9 00 65   5f
c0b0 : 58 85 58 60 38 a5 57 e9   cf
c0b8 : 03 85 57 a5 58 e9 00 85   e8
c0c0 : 58 60 38 a5 57 e9 f0 85   9f
c0c8 : 57 a5 58 e9 00 85 58 60   93
c0d0 : ac 1a c4 b1 57 c9 01 d0   5a
c0d8 : 03 ee 20 c4 60 a0 02 b1   69
c0e0 : 57 c9 01 f0 1f a9 01 a0   ff
c0e8 : 02 91 57 20 80 c0 ad 17   80
c0f0 : c4 ac 21 c4 91 59 c8 ad   4e
c0f8 : 18 c4 91 59 ee 1c c4 d0   86
c100 : 03 ee 1d c4 60 20 dd c0   5a
c108 : ee 17 c4 20 98 c0 20 dd   83
c110 : c0 ee 18 c4 20 a6 c0 20   60
c118 : dd c0 ce 17 c4 20 b4 c0   8d
c120 : 20 dd c0 ce 17 c4 20 b4   ba
c128 : c0 20 dd c0 ce 18 c4 20   89
c130 : c2 c0 20 dd c0 ce 18 c4   82
c138 : 20 c2 c0 20 dd c0 ee 17   bb
c140 : c4 20 98 c0 20 dd c0 ee   24
c148 : 17 c4 20 98 c0 20 dd c0   e2
c150 : ce 17 c4 ee 18 c4 20 b4   4a
c158 : c0 20 a6 c0 60 ce 1c c4   60
c160 : ad 1c c4 8d 1e c4 c9 ff   2d
c168 : d0 03 ce 1d c4 ad 1d c4   c9
c170 : 8d 1f c4 20 80 c0 ad 21   c9
c178 : c4 49 02 a8 b1 59 8d 17   c1
c180 : c4 c8 b1 59 8d 18 c4 20   2d
c188 : 24 c0 a9 00 8d 20 c4 a0   a5
c190 : 02 91 57 ee 17 c4 20 98   58
c198 : c0 20 d0 c0 ee 18 c4 20   b7
c1a0 : a6 c0 20 d0 c0 ce 17 c4   31
c1a8 : 20 b4 c0 20 d0 c0 ce 17   d3
c1b0 : c4 20 b4 c0 20 d0 c0 ce   f3
c1b8 : 18 c4 20 c2 c0 20 d0 c0   65
c1c0 : ce 18 c4 20 c2 c0 20 d0   24
c1c8 : c0 ee 17 c4 20 98 c0 20   68
c1d0 : d0 c0 ee 17 c4 20 98 c0   d0
c1d8 : 20 d0 c0 ce 17 c4 ee 18   ee
c1e0 : c4 20 a6 c0 20 b4 c0 ad   7c
c1e8 : 20 c4 c9 02 d0 10 ac 1a   91
c1f0 : c4 b1 57 aa 98 49 01 a8   e1
c1f8 : 8a 91 57 4c 12 c2 a2 00   6c
c200 : ad 20 c4 c9 03 d0 02 a2   2c
c208 : 01 ad 1a c4 49 01 a8 8a   53
c210 : 91 57 ce 1c c4 a9 ff cd   b9
c218 : 1c c4 d0 08 ce 1d c4 10   d4
c220 : 03 4c 27 c2 4c 73 c1 a9   26
c228 : 00 8d 1c c4 8d 1d c4 20   a4
c230 : 68 c0 ad 21 c4 49 02 a8   78
c238 : b1 59 8d 17 c4 c8 b1 59   e8
c240 : 8d 18 c4 20 24 c0 ad 1a   42
c248 : c4 49 01 a8 b1 57 8d 1b   48
c250 : c4 8d 14 c4 ac 1a c4 a9   7a
c258 : 00 91 57 ad 1b c4 c9 01   ad
c260 : d0 03 20 05 c1 ad 14 c4   be
c268 : c9 01 f0 0b a9 0f 8d 14   c1
c270 : c4 20 a5 c2 4c 7f c2 a9   25
c278 : 00 8d 14 c4 20 a5 c2 ce   b4
c280 : 1e c4 ad 1e c4 c9 ff d0   6c
c288 : 08 ce 1f c4 10 03 4c 94   cb
c290 : c2 4c 2f c2 ad 1a c4 49   ee
c298 : 01 8d 1a c4 ad 21 c4 49   08
c2a0 : 02 8d 21 c4 60 ad 17 c4   a3
c2a8 : 4a 8d 19 c4 ad 18 c4 4a   db
c2b0 : 85 5b a9 00 85 5a a9 28   6f
c2b8 : 85 59 20 00 c0 a9 04 18   8c
c2c0 : 65 58 18 ad 19 c4 65 57   09
c2c8 : 85 57 a9 04 65 58 85 58   c4
c2d0 : ad 17 c4 29 01 f0 11 ad   96
c2d8 : 18 c4 29 01 f0 05 a9 01   9d
c2e0 : 4c f6 c2 a9 04 4c f6 c2   91
c2e8 : ad 18 c4 29 01 f0 05 a9   f7
c2f0 : 02 4c f6 c2 a9 08 8d 15   69
c2f8 : c4 ad 14 c4 c9 ff f0 18   c1
c300 : a0 00 b1 57 aa bd 22 c4   a2
c308 : 4d 14 c4 0d 15 c4 4d 14   07
c310 : c4 aa bd 04 c4 91 57 60   10
c318 : a0 00 b1 57 aa bd 22 c4   ba
c320 : 2d 15 c4 f0 08 a9 01 8d   14
c328 : 16 c4 4c 32 c3 a9 00 8d   9e
c330 : 16 c4 60 a9 00 85 59 a9   da
c338 : 8e 85 5a a2 4e a0 df a9   30
c340 : 00 91 59 88 d0 f9 c6 5a   1d
c348 : ca 10 f6 a9 00 85 57 a9   ea
c350 : 60 85 58 a0 ef a2 4f a9   42
c358 : 01 91 57 88 88 88 ca 10   21
c360 : f8 a2 30 a0 02 a9 01 91   5e
c368 : 57 a0 ef 91 57 20 a6 c0   d0
c370 : ca 10 f0 a9 01 a0 ef a2   ce
c378 : 4f 91 57 88 88 88 ca 10   8f
c380 : f8 a2 00 a9 00 9d 22 c4   fe
c388 : ca d0 fa a2 0f bd 04 c4   46
c390 : a8 8a 99 22 c4 ca 10 f5   f7
c398 : a9 00 8d 21 c4 8d 1a c4   73
c3a0 : 8d 1c c4 8d 1d c4 60 20   d8
c3a8 : fd ae 20 9e b7 8e 17 c4   ae
c3b0 : 20 fd ae 20 9e b7 8e 18   91
c3b8 : c4 20 24 c0 a0 00 a9 01   60
c3c0 : 91 57 4c 05 c1 20 fd ae   23
c3c8 : 20 9e b7 8e 17 c4 20 fd   0b
c3d0 : ae 20 9e b7 8e 18 c4 20   2a
c3d8 : fd ae 20 9e b7 e0 01 f0   71
c3e0 : 09 e0 02 f0 0a a9 0f 4c   bb
c3e8 : f1 c3 a9 00 4c f1 c3 a9   dc
c3f0 : ff 8d 14 c4 4c a5 c2 78   41
c3f8 : 20 5d c1 58 60 78 20 5d   47
c400 : c1 4c fe c3 20 6c 7b 62   37
c408 : 7c e1 ff fe 7e 7f 61 fc   b8
c410 : e2 fb ec a0 00 ff 00 ff   3f
```

Listing 2. Benutzen Sie zur Eingabe des Maschinensprache-Teils von »Life« bitte den MSE







Tips & Tricks für Anfänger und Fortgeschrittene

Wiedereinmal können wir Ihnen eine Auswahl aus unserer Tips & Tricks-Kiste anbieten. Für fast jeden Leser dürften einige interessante Programmiertricks dabei sein.

Primzahlen-Einzeiler

Bei folgendem Einzeiler handelt es sich um ein Programm zur Errechnung und Ausgabe aller Primzahlen bis 1000. Es benötigt hierfür zirka 27 Sekunden.

```
10 DIMA(2000):FORI=2TO1000:IFA(I)=0THENPRINTI:
FORJ=I+ITO1000STEPI:A(J)=1:NEXTJ,I:01100 NEXT
```

Der Einzeiler ist zu lang, um mit ausgeschriebenen Basic-Befehlen eingegeben werden zu können. Man muß ihn also wie folgt eingeben:

```
10 DIMA(2000):FOI=2TO1000:IFA(I)=0TH?I:
FOJ=I+ITO1000STEPI:A(J)=1:NEJ,I:01100 NE
```

Das Programm ist in dieser Form noch nicht lauffähig. Man muß erst noch einen Befehl im Direktmodus eingeben, nämlich: POKE2109,0

Das Programm kann jetzt mit »RUN« gestartet und mit »SAVE« gespeichert werden. Es arbeitet nach dem »Sieb des Eratosthenes«, bei dem die Vielfachen einer Primzahl aus einer Liste gestrichen werden.

Im Programm steht 01100. Die erste 0 wird durch den POKE-Befehl überschrieben, dadurch denkt der Computer, daß die Zeile nach dem NEXTJ,I zu Ende ist. Die 1100 sind Pseudo-Zeilennummer und Koppeladresse, dadurch springt der Computer, wenn die IF/THEN-Bedingung nicht erfüllt ist, nicht zur nächsten Zeile, sondern zum zweiten NEXT. Sie können natürlich auch zwei oder drei Zeilen machen, um sich die Arbeit zu erleichtern.

(Michael Patra/tr)

Etiketten drucken mit dem MPS-801

Durch folgende Änderungen läßt sich das Programm »ETIKETTEN 64« im 4. Sonderheft, Seite 15, das in der Originalversion für einen Commodore-Drucker 1526 (MPS-802) geschrieben worden ist, auch mit einem MPS-801 (1525) betreiben. Das Problem bestand lediglich darin, den fest eingestellten Zeilenabstand des MPS-801 auf Null zu reduzieren, denn sonst hat der Ausdruck nicht viel Ähnlichkeit mit der erstellten Etikette auf dem Bildschirm.

Zeile 6930 wird gelöscht

```
7040   IFA$<>"←"THENPRINT#3,CHR$(15);A$;CHR$(8);:GOTO
       7080
7050   PRINT#3,CHR$(14);
7060   J=J+1:A$=MID$(T$,J,1)
7070   PRINT#3,A$;CHR$(15)CHR$(8);
```

Die eigentlichen Veränderungen stehen nur in den Zeilen 7040 und 7070.
CHR$(15) : Einschalten Standardzeichen
CHR$(8) : Einschalten des Bit-Modus

Der Trick besteht darin, daß, wenn sich der Drucker im Bit-Modus befindet, ein Zeilenvorschub dazu führt, daß die folgende Zeile ohne Zeilenabstand gedruckt wird und so der standardmäßige zirka 3-Pixelabstand entfällt. Dieses Prinzip ist allgemein anwendbar, zum Beispiel wenn man ein Bild der Grafik (mit Commodore-Grafikzeichen), so wie auf dem Bildschirm zu sehen, zu Papier bringen will. Die PRINT-AT-Zeilen für den Drucker müssen dann folgendes Format haben:
1000 print#lf,chr$(15);"...."chr$(8)
lf=logische Filenummer des OPEN-Befehls.

(Karsten Bluhm/tr)

Verbessertes PRINT USING mit Simons Basic

Die USE-Routine in Simons Basic vermag Zahlen mit einer definierten Anzahl an Vor- und Nachkommastellen auszugeben. So gibt zum Beispiel die Festlegung »USE "####.#####", STR$(A)« die Zahl A, die in einen String umgewandelt wird, mit vier Stellen vor und fünf Stellen hinter dem Dezimalpunkt aus. Im allgemeinen ein sehr nützlicher Befehl, wenn nicht folgender Nachteil auftreten würde: Diese Formatierungsroutine macht selbst vor Exponentialzahlen nicht Halt. So wird die Zahl 1.12345678E-05 mit der obigen Festlegung ausgegeben als "1.12345"! Sämtliche Dezimalen nach der fünften Stelle werden mitsamt dem Exponenten abgeschnitten! Selbst der Exponent wird wie Dezimalziffern nach dem Komma behandelt; es wird nicht nur die Mantisse formatiert!

Dieses Mißgeschick beseitigt die folgende Routine. Die zu formatierende Zahl wird als Variable A dem Unterprogramm übergeben. Das Unterprogramm ermittelt die Mantisse MA und den Zehnerexponenten EA der Zahl A.

```
50000   PROC FORMAT
50010   REM ZU FORMATIERENDE ZAHL : A
50030   LA=LOG(ABS(A))/LOG(10) : REM ZEHNERLOGARITH-
        MUS
50040   EA=INT((LA*10+.5)/10) : REM ZEHNEREXPONENT
50050   MA=A/10↑EA : REM MANTISSE
50100   END PROC
```

Aufgerufen wird die Routine mit »CALL FORMAT«. Die Mantisse MA und der Exponent EA können im Hauptprogramm an andere Variablen übergeben werden. Eine Möglichkeit wäre zum Beispiel bei vielen zu formatierenden Zahlen MA und EA in einem Feld (Matrix) zu speichern und nach Programmdurchlauf in einer Tabelle (mit Hilfe von USE) auszugeben.

Der Exponent EA der Zahl A kann in die wissenschaftliche Notation gebracht werden, in dem man ihn in ein Vielfaches von 3 umwandelt. Entsprechend muß die Mantisse MA umgerechnet werden (Multiplikation mit 10 beziehungsweise 100).

Dies geschieht wie folgt: Man dividiert EA durch 3 und erhält den Quotienten Q=EA/3. Der ganzzahlige Anteil (INT) mit 3 multipliziert ergibt den neuen Exponenten: EA=INT(Q)*3; der rationale Anteil (FRAC) multipliziert mit 3 ergibt den Faktor für die Mantisse: F=FRAC(Q)*3: MA=MA*1EF.
Beispiel: EA=11, angestrebt: EA=—9.
Q=EA/3=—3.666666666,
EA=INT(Q)*3=—3*3=—9,

F=FRAC(Q)*3=.666666666*3=2,
MA=MA*1EF=MA*1E2=MA*100.

Um Rundungsfehler zu vermeiden, sollte man bei der Berechnung des Faktors F auf ganzzahlige Werte runden (INT oder eigene Rundungsroutine, siehe weiter unten). Die Mantisse muß immer mindestens drei Stellen vor dem Komma besitzen, ein Vorzeichen und zwei Ziffern!

Um die Mantisse nicht einfach nach der kten Dezimalstelle abzuschneiden, sondern gerundet auszugeben, kann folgender Algorithmus verwendet werden:
INT((MA*1EK+.5)/1EK).

Liegt der Exponent EA als Vielfaches von 3 vor, so kann bei Meßergebnissen die Vorsilbe der Einheit (Kilo, Milli, Piko, etc.) ermittelt werden. Die entsprechende Vorsilbe kann dann als V$ der Maßeinheit (M$) vorangestellt werden (V$+M$). Die möglichen Vorsilben mit den dazugehörigen Zehnerexponenten sind:

Exa	E	10^{18},	Atto	a	10^{-18},
Peta	P	10^{15},	Femto	f	10^{-15},
Tera	T	10^{12},	Pico	p	10^{-12},
Giga	G	10^{9},	Nano	n	10^{-9},
Mega	M	10^{6},	Mikro	μ	10^{-6},
Kilo	k	10^{3},	Milli	m	10^{-3},
Hekto	h	10^{2},	Zenti	c	10^{-2},
Deka	d	10^{1},	Dezi	d	10^{-1}.

Wie man sieht, sind die Potenzen zu den Vorsilben Hekto, Deka, Zenti und Dezi keine Vielfachen von 3. Diese können mit Hilfe eines IF-THEN-Statements ausgesondert werden.
IF ABS(EA)>2 THEN EXEC VORSILBE,
wobei durch »VORSILBE« ein Unterprogramm aufgerufen wird, welches die Vorsilben zu den Exponenten, die Vielfache von 3 sind, ermittelt. Statt »EXEC« ist natürlich auch ein Sprung zu einer Marke (PROC...) mittels »CALL« möglich.

(Roland Wolff/tr)

Hilfe für/gegen Turbo-Tape

Wenn sie mit Turbo-Tape geladen wurden, stürzen manche Programme, die den Speicher von 50000 bis 51000 benutzen, plötzlich ab. Das liegt daran, daß Turbo-Tape den Vektor zum Holen eines Befehls auf eine eigene Routine richtet. Wird nun diese Routine überschrieben, steigt der Computer wenn er den nächsten (Basic-)Befehl sucht, aus, da das Programm, auf das der Vektor $308 zeigt, nicht mehr existiert. Die Veränderung dieses Vektors ist von Basic aus nicht möglich, aber durch »SYS 58451« (Standard Basic Vektoren $0300 bis $030B laden) wird er wieder »geradegebogen«.

(Julian Ziersch/tr)

Hilfe für Datasettenbesitzer

Kürzere Maschinenprogramme werden vom Programmierer gerne im Kassettenpuffer abgelegt. Den Floppy-Besitzer stört dies nicht. Er kann Programme laden und das Maschinenprogramm bleibt ihm erhalten. Nicht so beim Benutzer einer Datasette. Lädt er ein Programm von Kassette in den Bereich ab 828 ($033C), so wird es zerstört. Es gibt aber einen Trick, den Kassettenpuffer an einen anderen Speicherbereich zu legen. In den Speicherstellen 178 und 179 steht der Beginn des Kassettenpuffers. Normalerweise beginnt dieser bei 828. Durch Verändern dieses Vektors können wir den Kassettenpuffer beim Laden von Programmen schonen. Um ihn ans Ende des Speicherbereichs zu legen, schreiben Sie: POKE 179, PEEK (56)—2.

(Herbert Kunz/tr)

INPUT ohne Fragezeichen

Das manchmal lästige Fragezeichen beim Basic-Befehl »INPUT« kann man durch ein »POKE 19,1« verschwinden lassen. Allerdings sollte diese Manipulation sofort nach der INPUT-Anweisung durch »POKE 19,0« wieder rückgängig gemacht werden.

(Günther Stangl/tr)

Schachuhr

```
2 S=1—S:TI$=A$(S):FORI=1TO1:GETI:A$(S)=TI$:PRINT"
{CLR,3SPACE}?"A$(1),,,A$(0):IFA$(S)<E$THENNEXT:
GOTO2
```

Nachdem ich für ein Schachuhrprogramm eine Bildschirmseite verbraucht hatte, beschloß ich, dieses Programm ordentlich zu kürzen. Schließlich wurde es dann zu dem vorliegenden Einzeiler, der aus Ihrem Computer eine richtige Schachuhr macht.

```
A$(0)="000000":A$(1)="000000":E$="000020":
PRINT"{CLR}":GOTO 2
```

Zuerst müssen die Anfangszeiten der beiden Spieler definiert werden, die wohl jeweils Null sein dürften, wenn man nicht gerade gegen einen weitaus stärkeren Spieler spielt. (Jeweils A$(0) und A$C(1) im bekannten sechsstelligen Format der internen Uhr TI$). Danach definiert man die Endzeit E$ (wie oben), bei der das Programm stoppen soll und startet den Einzeiler nach einem SHIFT/CLR-HOME mit GOTO2 (nicht mit RUN, um die Stringvariablen nicht zu löschen).

Auf Druck der Zahlen 1-9 läuft jeweils die Uhr des Gegenspielers weiter. Will man eine Pause machen, drückt man irgendeinen Buchstaben.

Mit GOTO 2 kann das Programm dann fortgesetzt werden (ohne sich an dem »Syntax Error« zu stören), worauf die jeweils andere Uhr, als die, die vor Abbruch lief, weiterläuft.

Zur Erklärung des Einzeilers

S=1—S« ändert S jeweils von 0 auf 1 oder von 1 auf 0 (es läuft dann jeweils die Uhr von Spieler 1 oder 0). Nun wird die Uhr TI$ gestellt. Danach folgt eine For-Next-Schleife, die nur verlassen werden kann, wenn I eine der Zahlen von 1-9 annimmt, was nur bei Drücken irgendeiner Zahl geschehen kann. Sonst ist I durch die »GETI«-Abfrage immer Null. In der Schleife wird die jeweilige Zeit der beiden Spieler ausgegeben. Außerdem wird untersucht, ob die Endzeit erreicht ist.

(Oliver Stengelin/tr)

Der Hypra-Listschutz

Ich habe schon öfters im Leserforum gelesen, daß bei Aktivierung des Befehls »POKE 808,225« ein verstümmeltes Listing erscheint. Ich habe da eine Lösung gefunden: Nach Eingabe der Zahlenkombination 35072120 ist das Listing wieder normal sichtbar. Unglaublich, aber wahr!

Hier nun unsere Preisfrage an die Profis:
Was geschieht hier?

Der erste Leser, der uns die richtige Lösung schickt, bekommt einen 64'er-Aufkleber!

(Helmut Skolaut/tr)

Maschinenprogramme kopieren

Wenn man nicht zu den glücklichen Besitzern eines Kopierprogramms gehört, kann man Maschinenprogramme mit Hilfe des MSE kopieren: MSE laden — MSE starten — Programm laden — andere Diskette/Kassette einlegen — Programm mit CTRL-S wieder speichern. Basic-Programme lassen sich mit dieser Methode übrigens genauso kopieren.

(tr)

Geräusche von A bis Z

Durch Veränderung der Filterfrequenz und durch verschiedene Filter lassen sich einfache Geräusche erzeugen.
Zeile 10 S=54272:Rem Basisregister
Zeile 20 FORL=0TO24:POKES+L,0:NEXT

	Zeile 30	Zeile 40	Zeile 50	Zeile 60	Zeile 70	Zeile 80	Zeile 90	Zeile 100
Geräusch:	Frequenz	Hall	Grenzfrequenz	Resonanz	Pass	Wellenform	Schleife und POKE	Warteschleife und GOTO
Schuß	POKE S+0,0: POKE S+1,18	POKE S+5,1*16+11	POKE S+22,110	POKE S+23,15*16+3	POKE S+24,5*16+15	POKE S+4,0: POKE S+4,129	FORJ=1TO 255: POKE S+0,J: NEXT	FORA=1TO 1000: NEXT:GOTO80
Explosion	POKE S+0,0: POKE S+1,6	POKE S+5,2*16+13	POKE S+22,100	POKE S+23,15*16+3	POKE S+24,3*16+15	POKE S+4,0: POKE S+4,129	FORJ=1TO 100: POKE S+0,J: NEXT	FORA=1TO 4000: NEXT:GOTO80
Uhrenschlag	POKE S+0,0: POKE S+1,6	POKE S+5,1*16+10	POKE S+22,110	POKE S+23,15*16+3	POKE S+24,1*16+15	POKE S+4,0: POKE S+4,17	FORJ=1TO 255: POKE S+0,7: NEXT	FORA=1TO 500: NEXT:GOTO80
Brandung	POKE S+0,0: POKE S+1,40	POKE S+5,10*16+12	POKE S+22,0	POKE S+23,0	POKE S+24,0*16+15	POKE S+4,0: POKE S+4,129	FORJ=1TO 255: POKE S+0,J: NEXT	FORA=1TO 3500: NEXT:GOTO80

(Jürgen Hüsgen/tr)

Zum Beispiel:
Um einen Schuß zu erzeugen, muß das Programm wie folgt aussehen:

```
10  S=54272
20  FORL=0TO24:POKE S + L,0: NEXT
30  POKE S + 0,0 : POKE S + 1,18
40  POKE S + 5,1 * 16 + 11
50  POKE S + 22, 110
60  POKE S + 23, 15 * 16 + 3
70  POKE S + 24, 5 * 16 + 15
80  POKE S + 4, 0 : POKE S + 4, 129
90  FORJ = 1 TO 255 : POKE S + 0,J : NEXT
100  FORA = 1 T 1000 : NEXT : GOTO 80
```

POKE oder nicht POKE

Vielleicht ist manchen Computer-Fans, die öfters an unterschiedlichen C 64 arbeiten, schon folgendes aufgefallen: Bei einigen älteren Modellen bewirkt ein »POKE 1024,1« überhaupt nichts, bei den neueren hingegen erscheint in der oberen linken Bildschirmecke ein »A«. Warum erscheint bei den älteren Modellen das »A« nicht auf dem Bildschirm, obwohl doch alles korrekt gemacht wurde? Das liegt daran, daß das alte Betriebssystem des C 64 die Zeichen nicht in der Zeichen-, sondern in der Hintergrundfarbe in den Bildschirm schreibt. Sie sind also zwar vorhanden, aber nicht sichtbar. Dies werden sie erst, wenn durch einen entsprechenden Eintrag in den Farbspeicher ab Adresse 55296 die Farbe des Zeichens extra gesetzt wird. In unserem Fall mit dem »POKE 1024,1« müßte also noch ein »POKE 55296,14« (für ein hellblaues »A«) folgen. Deswegen sollten auch Besitzer eines neueren C 64, die uns Listings schicken, diese Farb-POKEs hinzufügen. Die Programme könnten sonst auf den älteren Computern merkwürdige Bildschirmdarstellungen haben. Für Besitzer eines EPROM-Brenners, die sich ein neues Kernel brennen wollen, sei noch kurz angemerkt, wie das Kernel »richtig« lauten muß, um den Farb-POKE automatisch zu setzen:
alt: $E4DA LDA $D021 (=Zeichen in Hintergrundfarbe)
neu: $E4DA LDA $0286 (=Zeichen in Zeichenfarbe)

(Frank Jahnke/tr)

?VERIFY ERROR

Da hat man ein gutes Basic-Programm geschrieben, und dann tritt beim obligatorischen VERIFY dieser Fehler auf. Aber in welcher Programmzeile steckt der Fehler? Dieses Programm (Listing 1) gibt die fehlerhafte Zeilen-Nummer an. Beide Programme müssen auf einer Diskette sein.
Folgende Fehleraussagen sind möglich:
1. Ungleiche Zeilenlänge
2. Zeichenfehler bei gleicher Zeilenlänge (oder ein Basic-Befehl ist anders)
3. Ungleiche Zeilennummer
4. Fehlen einer Zeile

Bei Fehlen einer Zeile beziehungsweise ungleicher Zeilennummer wird der Vergleich mit der Aussage »Bitte überprüfen« abgebrochen.

(Gerhard Reul/tr)

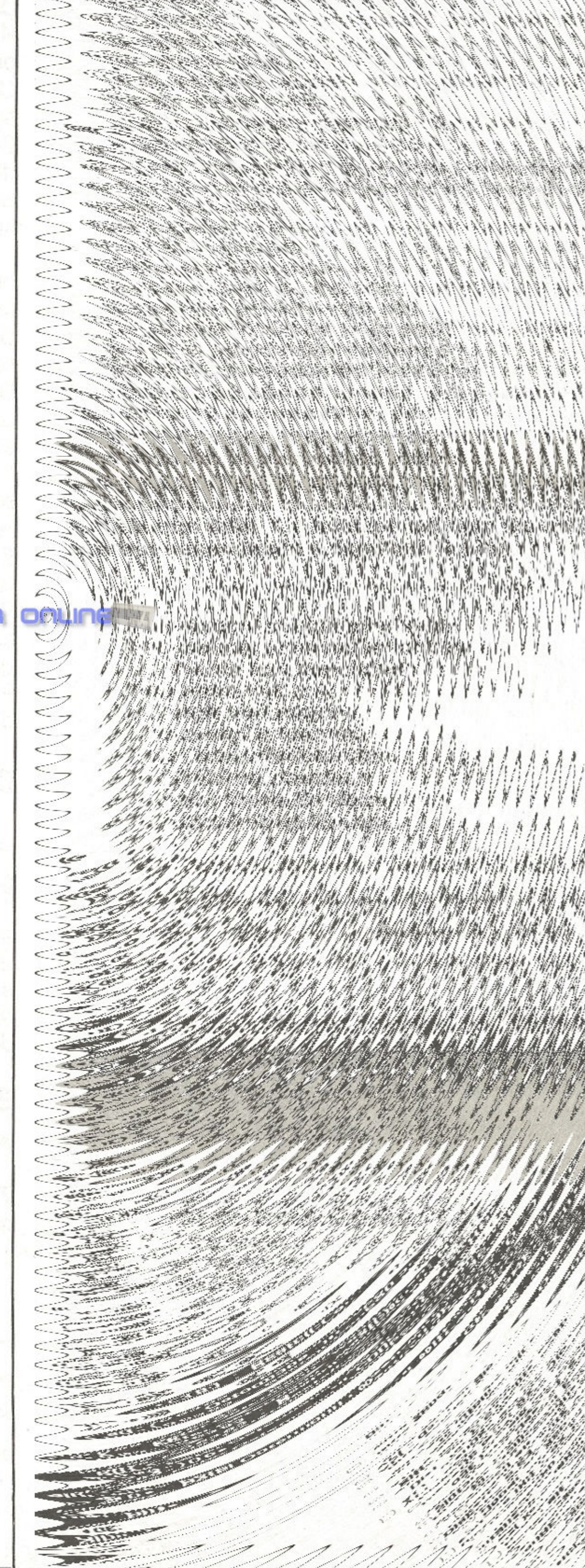
INT-Funktion fehlerhaft

Wie mir neulich auffiel, ist die »INT«-Funktion beim C 64 keineswegs die definitionsgemäße Integer-Funktion, wie man aus der Abkürzung entnehmen könnte, sondern die Gaussklammerfunktion. Glücklicherweise ist die Funktion im Handbuch richtig erklärt, nur stimmt nicht, daß »negative Zahlen dem Betrag nach größer werden«, wenn man die Nachkommastellen abschneidet. Die im Betriebssystem definierte Funktion rundet alle Zahlen ab, anstatt die Nachkommastellen abzuschneiden. Das heißt: $f(x) = [x]$ (Gaussklammer), und nicht: $f(x) = [|x|] \cdot sgn(x)$ (Integer). Aus INT(—1.23) wird nämlich —2 und nicht, wie es richtig heißen müßte, —1! Es ist also Vorsicht geboten bei Programmen oder Rechnungen, die die Integer-Funktion verlangen und bei denen x negativ ist. Bei kaufmännischen oder sonstigen Anwendungen des C 64, von denen nicht nur das Erfolgserlebnis des Programmierers abhängt, wäre es nötig, die Funktion im Programm neu zu definieren:
DEF FN INT(x) = INT (ABS(x)) * SGN(x)

(Matthias Möller/tr)

```
20 REM *************************               <154>
40 REM *  GERHARD REUL                *          <057>
50 REM *  8591 TROESTAU               *          <018>
60 REM *  TEL. 09232/1678             *          <163>
80 REM *************************               <214>
100 DIM X(256),Y(256):Z1$="......"             <058>
110 F1$="<> ZEILEN-LAENGE":F2$="ZEICHEN-FE
    HLER":F3$="<> ZEILEN-NR.!!!"               <039>
120 GOTO 450                                   <152>
130 OPEN 15,8,15,"I"                           <185>
140 OPEN 2,8,2,B1$+",P,R":F=1:GOSUB 590        <230>
150 OPEN 3,8,3,B2$+",P,R":F=2:GOSUB 590        <188>
160 GET#2,X$,X$                                <123>
170 GET#3,X$,X$                                <149>
180 REM====== LADEN PRG 1 ============         <202>
190 GET#2,X$:IF X$=""THEN X$=CHR$(0)           <068>
200 I=I+1:X(I)=ASC(X$)                         <153>
210 REM (ZU ZEILE 220)  ZEILEN-NR. BESTIMM
    EN                                         <031>
220 IF I=4 THEN Z=X(4)*256+X(3):Z$=RIGHT$(
    Z1$+STR$(Z),7):PRINT"{5SPACE}ZEILE"Z$"
    {2SPACE}OK{UP}"                            <059>
230 IF X(I)=0 THEN GOTO 260                    <195>
240 GOTO 190                                   <056>
250 REM  ZEILEN-ODER PRG.-ENDE BESTIMMEN       <248>
260 IF I=2 THEN:IF X(1)=0 AND X(2)=0 THEN
    CLOSE 2:CLOSE 3:CLOSE 15:GOTO 550:REM
    PRG.-ENDE                                  <149>
270 IF I<5 THEN GOTO 190                       <121>
280 II=I:I=0                                   <033>
290 REM====== LADEN PRG 2 ============         <062>
300 GET#3,Y$:IF Y$=""THEN Y$=CHR$(0)           <009>
310 J=J+1:Y(J)=ASC(Y$)                         <162>
320 IF Y(J)=0 THEN GOTO 340                    <047>
330 GOTO 300                                   <020>
340 IF J<5 THEN GOTO 300                       <183>
350 JJ=J:J=0                                   <020>
360 REM===== VERGLEICHEN =============         <237>
370 IF X(3)<>Y(3)OR X(4)<>Y(4)THEN:F$=F3$:
    GOSUB 420:GOTO 430                         <046>
380 IF II<>JJ THEN:F$=F1$:GOSUB 420:GOTO 1
    90                                         <082>
390 FOR I=5 TO II                              <191>
400 IF X(I)<>Y(I)THEN:F$=F2$:GOSUB 420:I=I
    I                                          <119>
410 NEXT:I=0:GOTO 190                          <218>
420 PRINT"{19RIGHT}"F$:RETURN                  <203>
430 PRINT"{DOWN,6SPACE,RVSON,3SPACE}BITTE
    UEBERPRUEFEN !{4SPACE,RVOFF}":CLOSE 2:
    CLOSE 3:CLOSE 15:GOTO 560                  <099>
440 REM======== MENUE =================        <074>
450 PRINT"{CLR,2DOWN,6SPACE,RVSON,SPACE}VE
    RGLEICHEN VON PROGRAMMEN{SPACE,RVOFF,2
    DOWN}"                                     <138>
460 PRINT"{2SPACE}ES KOENNEN ZWEI BASIC-PR
    OGRAMME,{DOWN}"                            <104>
470 PRINT"{2SPACE}DIE SICH AUF EINER DISKE
    TTE BEFINDEN,{DOWN}"                       <236>
480 PRINT"{2SPACE}MITEINANDER VERGLICHEN W
    ERDEN.{4DOWN}"                             <134>
490 INPUT"{3SPACE}PROGRAMM-NAME 1 ";B1$:PR
    INT:PRINT                                  <199>
500 INPUT"{3SPACE}PROGRAMM-NAME 2 ";B2$        <099>
510 PRINT"{CLR,2DOWN,5SPACE,RVSON,SPACE}VE
    RGLEICHEN VON PROGRAMMEN{SPACE,RVOFF,2
    DOWN}"                                     <198>
520 PRINT"{2SPACE}PROGRAMM 1:{9SPACE}PROGR
    AMM 2:{2DOWN}"                             <099>
530 PRINT"{2SPACE}"B1$"{16SPACE}"B2$"{2DOW
    N}":GOTO 130                               <037>
540 REM========= WARTEN ==============         <145>
550 PRINT"{2DOWN,13SPACE,RVSON,SPACE}ENDE{
    SPACE,RVSON}"                              <151>
560 GET A$:IF A$=""THEN 560                    <182>
570 RUN                                        <104>
580 REM====== FEHLER-KANAL ============         <166>
590 INPUT#15,A,A$:IF A=0 THEN RETURN           <207>
600 PRINT"{3SPACE}"F"{LEFT}. ";A$:CLOSE 2:
    CLOSE 3:CLOSE 15                           <218>
610 GOTO 560                                   <158>
```


Listing 1. »Vergleichen von Programmen«. Bitte benutzen Sie zur Eingabe den Checksummer.



Software-Hex-Tastatur

Das Programm (Listing 2) wird normal geladen und mit »RUN« gestartet. Es kopiert dann mit der schon im 64'er, Ausgabe 3/85, Seite 68 vorgestellten ROM-Routine das Kernel-ROM in das darunterliegende RAM (Zeilen 30 bis 40) und verändert dort die Tastatur-Decodiertabellen (Zeilen 50 bis 70). Wenn sich der Computer wieder mit »READY.« meldet (nach zirka 0,5 Sekunden), lädt man wie gewohnt den MSE, gibt »POKE 2096,53« ein und startet ihn mit »RUN«. Durch die Änderung der Speicherstelle 2096 wird beim MSE-Start der Wert 53 in die Speicherstelle 1 der Zeropage geschrieben. Deshalb holt sich der Computer seine Informationen im Kernel-Bereich nicht mehr aus dem ROM, sondern aus dem RAM, wo die geänderte Tastaturdecodiertabelle steht. Die neue Tastenbelegung wird wie folgt aktiviert: Die normalen Funktionen bleiben bestehen, mit »Shift« ergeben sich aber folgende Umbelegungen:
— auf M bis / liegen 0 bis 3
— auf K bis ; liegen 4 bis 7
— auf I bis @ liegen 8 bis B
— auf 9 bis — liegen C bis F.

Auch die Belegung der INST-DEL-Taste wurde verändert (Zeile 70): ohne Shift wird jetzt INST, mit Shift DEL ausgeführt. Es empfiehlt sich also, bei der Eingabe von MSE-Programmen die Shift-Lock-Taste zu drücken und dann mit den neuen Tasten zu arbeiten. Als Merkhilfe kann man seitlich auf die Tasten kleine Aufkleber mit der neuen Belegung kleben.

(Sven Heemeyer/Andreas Meyer/tr)

```
10 REM MSE-HEX-TASTATUR                          <195>
20 REM VON S. HEEMEYER & A. MEYER                <185>
30 S=820:FOR I=S TO S+6:READ A:POKE I,A:NE
   XT I                                          <178>
40 POKE 88,0:POKE 89,0:POKE 90,0:POKE 91,0
   :POKE 780,0:POKE 781,224:SYS S                <016>
50 FOR I=48 TO 57:READ A:POKE 60354+A,I:NE
   XT                                            <128>
60 FOR I=65 TO 70:READ A:POKE 60354+A,I:NE
   XT                                            <253>
70 POKE 60289,148:POKE 60354,20                  <152>
80 PRINT"{CLR}MSE LADEN, DANN POKE2096,53:
   RUN"                                          <047>
90 DATA 133,95,134,96,76,191,163                 <036>
100 DATA 36,47,44,55,37,42,45,50,33,38           <022>
110 DATA 41,46,32,35,40,43                       <135>
```

Listing 2. »Hex-Tastatur«.

Die unmögliche Uhr

Dieses Programm stellt während des Editierens oder Ablaufs von Basic- oder Maschinenspracheprogrammen im unteren Rand des Bildschirms eine absolut genaue Uhr mit Stunden-, Minuten und Sekundenanzeige dar. Es verbraucht keinen Basic-Speicherplatz, da das Steuerprogramm von $C000 bis $C2D7 und die Spritedaten unter dem Kernel-ROM liegen. Es verfügt über einen minutengenau einstellbaren Wecker. Zur Funktionsweise des Programmes:

Der VIC II-Chip wird in der IRQ-Routine veranlaßt, an zwei Bildschirmstellen einen Interrupt auszulösen. Die erste dieser Stellen liegt am unteren Bildschirmrand. Hier wird durch Ändern der Speicherstelle 53265 der Bildschirm »verlängert«. Da die Uhr unabhängig von Programmen laufen soll, in denen auch Sprites vorkommen, werden die aktuellen Spritedaten gerettet und erst dann die Daten für die Uhr in die entsprechenden VIC-Register geschrieben. Zusätzlich wird die Bank-select-Speicherstelle 56576 »umgeschaltet«, um die Uhr-Sprite-Daten unter dem Kernel-ROM lesen zu können. Anschließend werden die Spritezeiger nach der exakten, weil durch Netzfrequenz getakteten Tageszeituhr des CIA 2-Chip korrigiert und bei Erreichen der Alarmzeit ein optisches und akustisches Signal ausgegeben.

Die zweite Bildschirmstelle, an der ein Interrupt ausgelöst wird, liegt im unteren, nicht mehr sichtbaren Bildschirmbereich. Hier erhalten die VIC-Register und die Bank-select-Speicherstelle wieder ihre ursprünglichen Werte.

Nun zur Bedienung des Programms:

Nachdem Sie das Listing 3, »BORDER-CLOCK«, das Sie mit dem MSE abgetippt und gespeichert, gestartet haben, erzeugt dieses erst das eigentliche Programm »UHR« auf der Diskette. Dies hat folgenden Grund: Das endgültige Programm »UHR« enthält die vollständigen Daten für die Sprite-Ziffern 0 bis 9 und den Doppelpunkt. Diese Daten werden in »BORDER-CLOCK« erst aus den sieben Segmenten einer Digital-Uhr zusammengesetzt. Dies bewirkt, daß »BORDER-CLOCK« etwa 200 Byte kürzer ist als »UHR«. Nach dem Laden und Starten des »UHR«-Programms müssen Zeit und Weckzeit in TI$-Schreibweise (je zwei Stellen für Stunden, Minuten und Sekunden) eingegeben werden.

Nach Drücken von RUN/STOP-RESTORE muß die Anzeige mit »SYS 49756« neu aktiviert werden, die Uhr läuft jedoch intern weiter. Vor der Arbeit mit Peripheriegeräten (Floppy, Drucker, etc.) muß die Anzeige ausgeschaltet werden. Dies kann durch die Tastenkombination CTRL-RESTORE erfolgen. Danach kann sie durch Commodore-Taste und RESTORE wieder eingeschaltet werden. Am Programm können Sie natürlich selbst einiges verändern.

Die Sprite-Daten der Ziffern 0 bis 9 und des Doppelpunktes liegen von 3270 bis 3974. Die Farbe des Uhrenhintergrundes ist in 2974 enthalten, die Farben der einzelnen Sprites von 2980 bis einschließlich 2987. Es können auch einzelne Ziffern ausgeschaltet bleiben, und zwar durch POKE 2962, PEEK(2962)AND(255-2↑I), wobei I zwischen 0 und 7 die auszuschaltende Stelle angibt. Schließlich kann noch die Frequenz des Alarmgongs mittels POKE 3072,INT(F/256): POKE 3067,F-256*PEEK(3072) mit F als Frequenz geändert werden.

(Anton Ernst/tr)

Das 64'er EPROM-Programmiergerät

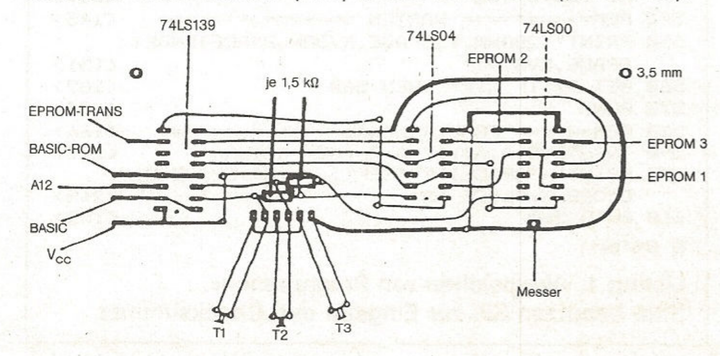
In der Testphase unseres Programmiergerätes haben sich zwei Veränderungen am Schaltplan des 64'er EPROM-Programmiergerätes ergeben (Ausgabe 12/85, Seite 47, Bild 5).
1. Auf der PGM-Leitung muß ein 7404-Inverter zwischengeschaltet werden.
2. Die vom 74LS139 (Pin 7) ausgehende Leitung muß zwischen dem Treiber und dem Invertierer abgezweigt und über eine Diode (1N4148) mit der Leitung zum Umschalter verbunden werden.

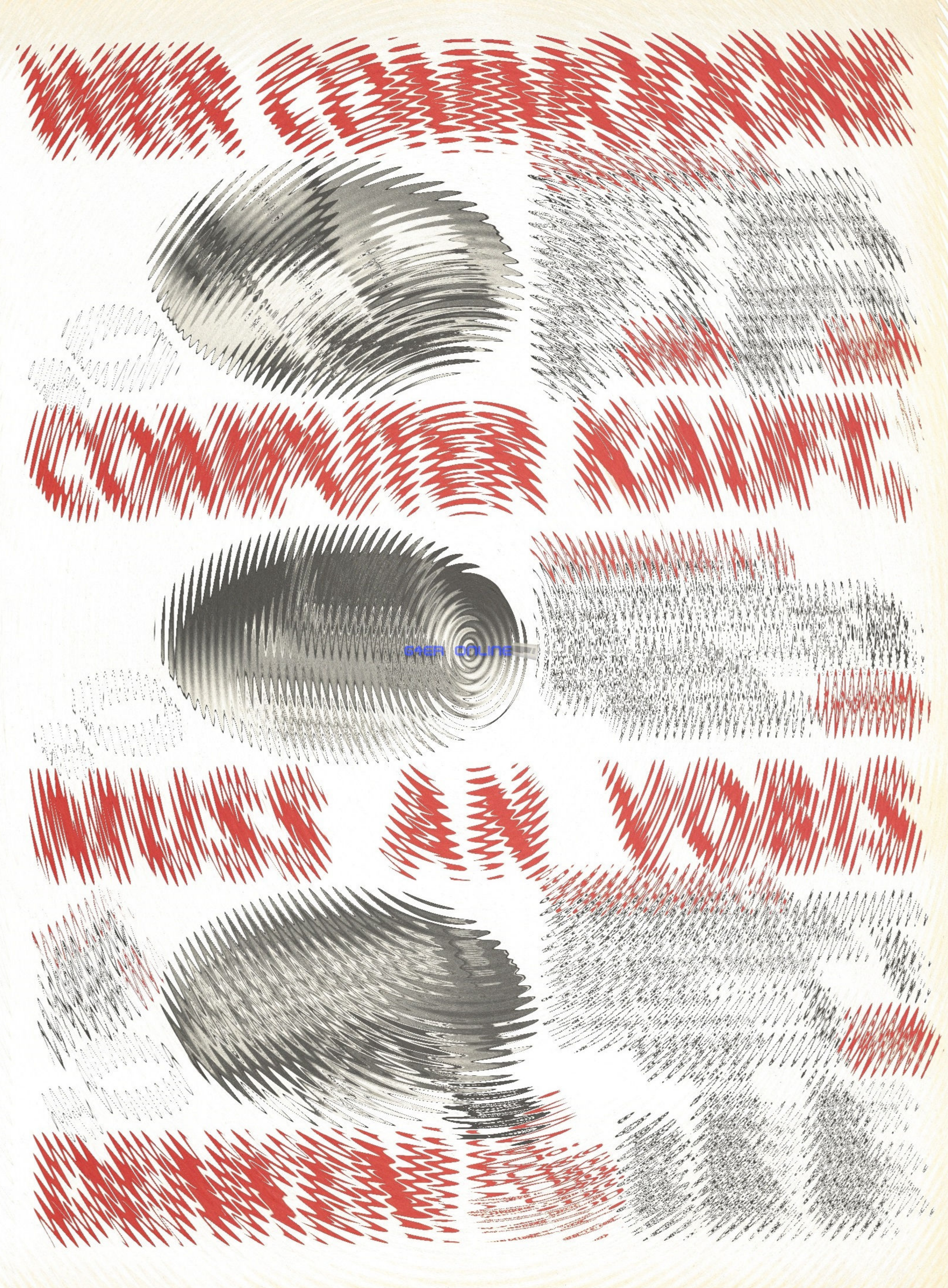
(aw)

EPROM-Trans

Ausgabe 10/85, Seite 45, Bild 4

Bei der Zusatzplatine müssen die beiden ICs 74LS00 und 74LS04 vertauscht werden. Taster T1 und T2 müssen noch mit zwei Widerständen (1k5) bestückt werden (siehe Bild).





```
programm : border-clock   0801 0efb
-----------------------------------
0801 : 26 08 0a 00 9f 32 2c 38   5b
0809 : 2c 32 2c 22 40 30 3a 55   b7
0811 : 48 52 2c 50 2c 57 22 3a   12
0819 : 98 32 2c c7 28 31 29 c7   0f
0821 : 28 38 29 3b 00 47 08 14   9a
0829 : 00 81 49 b2 30 a4 31 32   e4
0831 : 32 30 3a 98 32 2c c7 28   11
0839 : c2 28 32 34 30 36 aa 49   14
0841 : 29 29 3b 3a 82 00 5a 08   b7
0849 : 64 00 47 b2 33 37 30 34   ec
0851 : 3a 86 46 28 38 30 30 29   7d
0859 : 00 98 08 6e 00 81 49 b2   0c
0861 : 30 a4 36 3a 81 59 b2 31   c8
0869 : a4 c2 28 33 36 39 37 aa   3e
0871 : 49 29 3a 42 28 49 2c 59   56
0879 : 29 b2 c2 28 47 29 3a 47   e6
0881 : b2 47 aa 31 3a 57 28 49   39
0889 : 2c 59 29 b2 c2 28 47 29   df
0891 : 3a 47 b2 47 aa 31 00 be   b6
0899 : 08 78 00 82 59 2c 49 3a   be
08a1 : 81 58 b2 30 a4 39 3a 81   01
08a9 : 59 b2 30 a4 36 3a 81 5a   ec
08b1 : b2 31 a4 c2 28 33 36 39   e5
08b9 : 37 aa 59 29 00 f2 08 82   7e
08c1 : 00 8b c2 28 33 36 32 37   59
08c9 : aa 37 ac 58 aa 59 29 b2   c4
08d1 : 31 a7 41 b2 42 28 59 2c   a0
08d9 : 5a 29 aa 58 ac 36 34 3a   3f
08e1 : 46 28 41 29 b2 46 28 41   31
08e9 : 29 b0 57 28 59 2c 5a 29   f8
08f1 : 00 18 09 8c 00 82 5a 2c   a7
08f9 : 59 2c 58 3a 81 49 b2 30   53
0901 : a4 31 31 ac 36 34 3a 97   3d
0909 : 35 37 33 34 34 aa 49 2c   43
0911 : 46 28 49 29 3a 82 00 44   23
0919 : 09 96 00 41 b2 36 30 3a   a8
0921 : 81 49 b2 30 a4 32 3a 46   4b
0929 : 28 36 35 39 aa 33 ac 49   6a
0931 : 29 b2 41 3a 46 28 36 37   38
0939 : 34 aa 33 ac 49 29 b2 41   50
0941 : 3a 82 00 64 09 a0 00 81   e2
0949 : 49 b2 30 a4 31 31 ac 36   48
0951 : 34 3a 98 32 2c c7 28 46   3d
0959 : 28 49 29 29 3b 3a 82 3a   99
0961 : a0 32 00 00 00 1a 08 0a   20
0969 : 00 97 20 35 33 32 38 30   ea
0971 : 2c 30 32 3a 97 20 35 33   3f
0979 : 32 38 31 2c 32 00 46 08   e6
0981 : 14 00 99 22 93 11 11 11   68
0989 : 1d 9e 55 48 52 5a 45 49   f3
0991 : 54 3a 20 20 20 20 05 22   6a
0999 : 3b 3a 9f 31 2c 30 2c 30   55
09a1 : 3a 84 31 2c 54 24 3a a0   80
09a9 : 31 00 76 08 1e 00 99 c7   51
09b1 : 28 31 33 29 20 22 11 11   dd
09b9 : 1d 9b 57 45 43 4b 5a 45   a5
09c1 : 49 54 3a 20 20 20 05 22   22
09c9 : 3b 3a 9f 31 2c 30 2c 30   85
09d1 : 3a 84 31 2c 57 24 3a a0   e0
09d9 : 31 00 9d 08 28 00 48 24   5f
09e1 : b2 c8 28 54 24 2c 32 29   4b
09e9 : 3a 4d 24 b2 ca 28 54 24   b1
09f1 : 2c 33 2c 32 29 3a 53 24   02
09f9 : b2 c9 28 54 24 2c 32 29   e3
0a01 : 00 c7 08 32 00 42 b2 30   6a
0a09 : 3a 8b c5 28 48 24 29 b1   2d
0a11 : 31 32 a7 42 b2 31 3a 48   bc
0a19 : 24 b2 c9 28 c4 28 c5 28   03
0a21 : 48 24 29 ab 31 32 29 2c   dd
0a29 : 32 29 00 d9 08 3c 00 8b   a5
0a31 : 20 48 24 b2 22 30 30 22   7d
0a39 : a7 42 b2 31 00 04 09 46   a5
0a41 : 00 97 20 35 36 35 38 37   18
0a49 : 2c 31 36 ac c5 28 c8 28   42
0a51 : 48 24 2c 31 29 29 aa c5   ef
0a59 : 28 c9 28 48 24 2c 31 29   34
0a61 : 29 b0 42 ac 31 32 38 00   8e
0a69 : 29 09 50 00 97 20 35 36   e7
0a71 : 35 38 36 2c 31 36 ac c5   d8
0a79 : 28 c8 28 4d 24 2c 31 29   74
0a81 : 29 aa c5 28 c9 28 4d 24   d1
0a89 : 2c 31 29 29 00 4e 09 5a   09
0a91 : 00 97 20 35 36 35 38 35   64
0a99 : 2c 31 36 ac c5 28 c8 28   92
0aa1 : 53 24 2c 31 29 29 aa c5   4a
0aa9 : 28 c9 28 53 24 2c 31 29   e5
0ab1 : 29 00 7a 09 64 00 81 20   27
0ab9 : 49 b2 30 20 a4 20 33 3a   f8
0ac1 : 97 35 31 31 30 30 aa 49   27
0ac9 : 2c c5 28 ca 28 57 24 2c   61
0ad1 : 49 aa 31 2c 31 29 29 aa   98
0ad9 : 31 32 38 3a 82 00 8f 09   f1
0ae1 : 6e 00 97 20 35 36 35 38   84
0ae9 : 34 2c 30 3a 99 22 11 11   98
0af1 : 11 22 00 9a 09 78 00 9e   f8
0af9 : 20 32 34 36 30 00 00 00   09
0b01 : a9 ef 85 ae a9 09 85 af   31
0b09 : a9 00 85 b0 a9 c0 85 b1   44
0b11 : a0 00 b1 ae 91 b0 e6 ae   8b
0b19 : d0 02 e6 af e6 b0 d0 02   d5
0b21 : e6 b1 a5 b1 c9 c3 d0 ea   53
0b29 : a9 c6 85 ae a9 0c 85 af   dd
0b31 : a9 00 85 b0 a9 e0 85 b1   6d
0b39 : a0 00 b1 ae 91 b0 e6 ae   b3
0b41 : d0 02 e6 af e6 b0 d0 02   fd
0b49 : e6 b1 a5 b1 c9 e5 d0 ea   8c
0b51 : 4c 5c c2 78 a9 7f 8d 0d   72
0b59 : dc a9 2c 8d 14 03 a9 c0   48
0b61 : 8d 15 03 a9 01 8d 1a d0   f5
0b69 : a9 f7 8d 12 d0 a2 2f bd   0e
0b71 : 00 d0 9d 5c c1 ca 10 f7   6f
0b79 : a9 00 8d 08 dd 58 60 78   9a
0b81 : ad 19 d0 8d 19 d0 ad 12   94
0b89 : d0 c9 f7 d0 2f a2 18 bd   3a
0b91 : 00 d0 9d 5c c1 bd 8e c1   b4
0b99 : 9d 00 d0 ca 10 f1 a9 c4   84
0ba1 : 8d 00 dd a9 00 8d 12 d0   31
0ba9 : a2 13 bd 1b d0 9d 77 c1   03
0bb1 : bd a9 c1 9d 1b d0 ca 10   ea
0bb9 : f1 4c 90 c0 20 ca c1 a2   b1
0bc1 : 18 bd 5c c1 9d 00 d0 ca   ba
0bc9 : 10 f7 a9 f7 8d 12 d0 a2   30
0bd1 : 13 bd 77 c1 9d 1b d0 ca   64
0bd9 : 10 f7 a9 97 8d 00 dd 58   43
0be1 : 4c 31 ea ea a0 00 ad 0b   b5
0be9 : dd 29 10 18 4a 4a 4a 4a   17
0bf1 : 69 80 48 ad 0b dd c9 12   4d
0bf9 : d0 09 a9 80 8d f8 c7 68   59
0c01 : 4c c1 c0 c9 92 d0 09 a9   be
0c09 : 81 8d f8 c7 68 4c c1 c0   f9
0c11 : 68 8d f8 c7 ad 0b dd 29   74
0c19 : 0f 69 80 48 ad 0b dd c9   44
0c21 : 12 d0 09 a9 80 8d f9 c7   ff
0c29 : 68 4c ea c0 c9 92 d0 09   10
0c31 : a9 82 8d f9 c7 68 4c ea   85
0c39 : c0 68 20 32 c1 a9 8a 8d   2a
0c41 : fa c7 ad 0a dd 29 f0 4a   4b
0c49 : 4a 4a 4a 69 80 8d fb c7   6c
0c51 : ad 0a dd 29 0f 69 80 8d   f9
0c59 : fc c7 a9 8a 8d fd c7 ad   38
0c61 : 09 dd 29 f0 4a 4a 4a 4a   76
0c69 : 69 80 8d fe c7 ad 09 dd   1f
0c71 : 29 0f 69 80 8d ff c7 ad   df
0c79 : 0e dd 09 80 8d 0e dd ad   e4
0c81 : 08 dd 4c 57 c1 8d f9 c7   76
0c89 : ad 0b dd e9 80 10 01 60   be
0c91 : ee f8 c7 ee f9 c7 ee f9   58
0c99 : c7 ad f9 c7 e9 8a 10 01   e3
0ca1 : 60 69 7f 8d f9 c7 ee f8   d3
0ca9 : c7 60 4c da c1 32 60 00   3e
0cb1 : 00 00 00 00 00 00 00 00   b2
0cb9 : 00 00 00 00 00 00 00 00   ba
0cc1 : 1b fa d1 00 00 c8 00 15   3e
0cc9 : f9 f1 00 00 00 00 00 fa   b1
0cd1 : f2 f1 f2 f3 f4 f0 f1 f2   7b
0cd9 : f3 f4 f5 f6 f7 fc ff ea   e0
0ce1 : ea 32 0b 55 0b 73 0b 91   ed
0ce9 : 0b b4 0b d2 0b f0 0b 14   f8
0cf1 : 0b 80 17 59 9a 36 ff c8   1a
0cf9 : 00 15 79 f0 ff 00 00 00   00
0d01 : 00 fa 0c f1 f2 f3 f4 f0   44
0d09 : 01 01 01 01 01 01 01 01   09
0d11 : 02 ff 07 a2 18 bd 00 d0   3a
0d19 : dd 8e c1 d0 06 ca 10 f5   ab
0d21 : 60 ea ea e0 11 f0 f6 9d   7d
0d29 : 5c c1 4c 66 c1 ad 7c c1   45
0d31 : 8d ae c1 a2 01 bd 9c c7   da
0d39 : dd f8 c7 d0 12 bd 9e c7   b7
0d41 : dd fb c7 d0 0a 4c cb c2   df
0d49 : ad 88 c7 c9 0a 10 03 4c   2b
0d51 : 81 ea ad 87 c7 c9 d3 f0   a0
0d59 : 2d a9 0f 8d 18 d4 a9 00   9f
0d61 : 8d 00 d4 a9 0c 8d 01 d4   33
0d69 : a9 0c 8d 05 d4 a9 00 8d   d2
0d71 : 06 d4 a9 11 8d 04 d4 a9   0d
0d79 : d3 8d 87 c7 a9 00 8d 7e   bb
0d81 : c7 a9 28 8d 7f c7 a9 00   b5
0d89 : 8d 88 c7 ee 89 c7 ad 89   cb
0d91 : c7 c9 05 d0 17 a9 00 8d   72
0d99 : 04 d4 a9 11 8d 04 d4 a9   33
0da1 : 00 8d 89 c7 ad af c1 49   b5
0da9 : 08 8d af c1 4c 81 ea 20   59
0db1 : 00 c0 a9 88 8d 18 03 a9   86
0db9 : c2 8d 19 03 a9 00 8d ff   b9
0dc1 : 3f 8d ff ff 8d 87 c7 8d   16
0dc9 : 88 c7 4c 85 c2 4b 52 4f   67
0dd1 : 57 20 4b 43 4f 4c 43 0d   f2
0dd9 : 4c 83 a4 08 48 8a 48 98   3c
0de1 : 48 ad 8d 02 c9 02 d0 0d   ad
0de9 : a9 2c 8d 14 03 a9 c0 8d   2a
0df1 : 15 03 4c b0 c2 c9 04 d0   dd
0df9 : 0a a9 b9 8d 14 03 a9 c2   7d
0e01 : 8d 15 03 68 a8 68 aa 68   30
0e09 : 28 4c 47 fe 78 ad 19 d0   04
0e11 : 8d 19 d0 ad 12 d0 f0 03   86
0e19 : 4c 31 ea 4c 81 ea ca 10   fd
0e21 : 06 ee 88 c7 4c f5 c1 4c   cd
0e29 : e2 c1 01 01 01 00 01 01   62
0e31 : 01 00 00 01 00 00 01 00   57
0e39 : 01 00 01 01 01 00 01 01   b1
0e41 : 00 01 01 00 01 01 00 01   1c
0e49 : 01 01 00 01 00 01 01 00   f7
0e51 : 01 00 01 01 01 01 00 01   cd
0e59 : 01 01 01 01 00 01 00 00   43
0e61 : 01 00 01 01 01 01 01 01   e1
0e69 : 01 01 01 01 01 00 01 01   61
0e71 : 09 0a 0a 09 09 09 09 00   20
0e79 : 3f 01 ff 02 fc 03 1f 04   e5
0e81 : ff 05 f8 06 0f 07 ff 08   3b
0e89 : f0 00 80 03 c0 06 e0 09   cc
0e91 : e0 0c e0 0f e0 12 e0 15   de
0e99 : e0 18 c0 1b 80 02 01 05   3f
0ea1 : 03 08 07 0b 07 0e 07 11   eb
0ea9 : 07 14 07 17 07 1a 03 1d   e7
0eb1 : 01 1b 3f 1c ff 1d fc 1e   ac
0eb9 : 7f 1f ff 20 fe 21 3f 22   06
0ec1 : ff 23 fc 21 80 24 c0 27   30
0ec9 : e0 2a e0 2d e0 30 e0 33   16
0ed1 : e0 36 c0 39 80 23 01 26   95
0ed9 : 03 29 07 2c 07 2f 07 32   23
0ee1 : 07 35 07 38 03 3b 01 33   c0
0ee9 : 0f 34 ff 35 f0 36 1f 37   64
0ef1 : ff 38 f8 39 3f 3a ff 3b   ae
0ef9 : fc 42 d0 0b 88 b1 45 99   0a
```

Listing 3. Die »unmögliche Uhr« müssen Sie mit dem MSE eingeben.

Der Riesen-Bildschirm

Dieses Hilfsprogramm (Listing 4) können Sie beim Schreiben eigener Programme verwenden, um Sprites auch über den oberen und unteren Rand hinaus zu bewegen. Das Programm kann von Basic oder Maschinensprache aus benutzt werden, da es im Kassettenpuffer von dezimal 828 bis 929 liegt und so den freien Speicherbereich nicht einschränkt. Die Funktionsweise des Programms:

Der VIC II-Chip löst an drei Bildschirmstellen einen Interrupt aus. Die erste liegt am unteren Bildschirmrand, hier wird durch einen Programmiertrick der Bildschirm »verlängert«. Die zweite Stelle liegt sieben Rasterzeilen unterhalb der ersten. Hier wird durch Auswertung der Speicherstelle 931 entschieden, ob und welche Sprites im unteren Randbereich sichtbar sein sollen. Die dritte Stelle liegt im unteren, nicht mehr sichtbaren Bildbereich, wo abhängig vom Inhalt der Speicherstelle 930 Sprites vom Anfang des oberen bis zum Anfang des unteren Randbereichs sichtbar gemacht werden.

Zur Bedienung des Programms:

Nach der Initialisierung mittels SYS 828 werden Sprites nicht mehr durch POKE 53269,X ein- oder ausgeschaltet, sondern durch POKE 930,X1 und POKE 931,X2. Das Bitmuster (!) von X1 gibt an, welche Sprites Y-Koordinaten von 0 bis 255 haben, X2 enthält das Bitmuster der Sprites mit Y-Koordinaten über 255. Ansonsten werden die Y-Koordinaten in den normalen Registern von 53249 bis 53263 eingetragen. Auf diese Weise sind Y-Koordinaten von 0 bis 511 möglich, wobei die Sichtbarkeit der Sprites je nach vorhandenem Monitor oder Fernsehgerät variiert.

Normalerweise sind Sprites mit Y-Koordinaten von 290 gerade noch sichtbar.

Hierzu einige Beispiele:

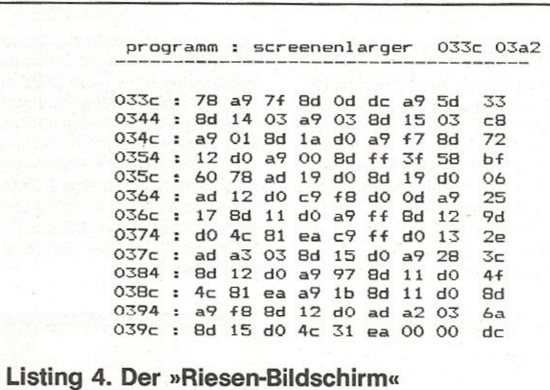
```
POKE  930,PEEK(930) OR 2↑3:POKE
      931,PEEK(931)AND(255—2↑3):
POKE  53249+2*3,5 — Sprite 4 hat die Y-Koordinate 5
POKE  930,PEEK(930)AND(255—2↑7):POKE 931,PEEK(931)
      OR 2↑7:
POKE  53249+7*2,7 — Sprite 8 hat die Y-Koordinate 262
POKE  930,PEEK(930)OR2↑0:POKE 931,PEEK(931)OR2↑0:
POKE  53249,3 — Sprite 1 hat die Y-Koordinaten 3 und 258, ist
      also zweimal auf dem Bildschirm sichtbar.
```

So ist eine Darstellung von 16 Sprites möglich. Das Demoprogramm im Listing 5 bewegt ein Sprite über den erweiterten Bildschirm.

Noch ein Tip zur Verwendung dieses Utilitys:

Beim Einbau in eigene Programme sollten Sie darauf achten, daß die letzte vom VIC II-Chip addressierbare Speicherstelle, also im Normalfall 16383, den Wert 0 besitzt, da sonst schwarze Streifen Ihren Bildrand verunstalten.

(Anton Ernst/tr)



```
 programm : screenenlarger   033c 03a2
---------------------------------------

033c : 78 a9 7f 8d 0d dc a9 5d  33
0344 : 8d 14 03 a9 03 8d 15 03  c8
034c : a9 01 8d 1a d0 a9 f7 8d  72
0354 : 12 d0 a9 00 8d ff 3f 58  bf
035c : 60 78 ad 19 d0 8d 19 d0  06
0364 : ad 12 d0 c9 f8 d0 0d a9  25
036c : 17 8d 11 d0 a9 ff 8d 12  9d
0374 : d0 4c 81 ea c9 ff d0 13  2e
037c : ad a3 03 8d 15 d0 a9 28  3c
0384 : 8d 12 d0 a9 97 8d 11 d0  4f
038c : 4c 81 ea a9 1b 8d 11 d0  8d
0394 : a9 f8 8d 12 d0 ad a2 03  6a
039c : 8d 15 d0 4c 31 ea 00 00  dc
```

Listing 4. Der »Riesen-Bildschirm«

```
1 POKE 53248,100:POKE 53287,7:POKE 2040,11
  :FOR I=0 TO 63:POKE 704+I,255:NEXT          <046>
2 FOR I=0 TO 290:POKE 53249,I+255*(I>255):
  POKE 930,-(I<256):POKE 931,-(I>255):NEXT
  :GOTO 2                                     <074>
```

© 64'er

Listing 5. »Demo Enlarger«. Bitte verwenden Sie zur Eingabe den Checksummer.

Die Commander-Datasette streikt nie wieder

Der Artikel über die Datasettenverbesserung hat bei mir und auch bekannten Computer-Fans Anklang gefunden. Leider mußte ich bald darauf feststellen, daß die Einbauanleitung des Komperators nur für die neue Datasette ausgelegt war. Ein Computerkollege, der leider nur eine Datasette des Typs »Commander — Computer Datasette Model NO: PM-4401C« hat, und den Komperator einbauen wollte, stieß dabei auf diese Schwierigkeiten und bat mich als Elektroniker um Abhilfe. Nach detaillierten Messungen mit dem Oszilloskop fand ich dabei die Anschlüsse heraus und baute den Komperator in seiner Datasette ein. Um auch anderen Computerfans die Möglichkeit zu geben, sich die Schaltung in ihre Commander-Datasette einzubauen, zeigen die beiden Pfeile in Bild 1 den Anschluß an die Platine.

(Peter Ehlert/tr)

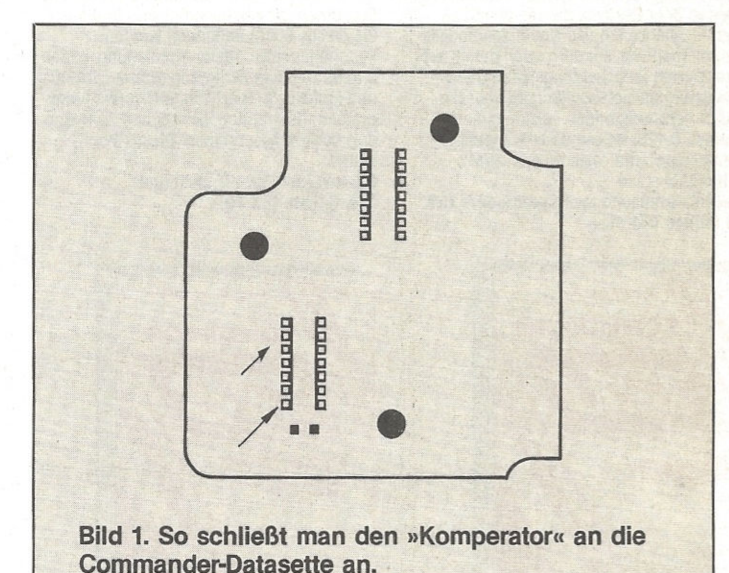


Bild 1. So schließt man den »Komperator« an die Commander-Datasette an.



Fehlerteufelchen

Musik-Kurs, Ausgabe 11/85, Seite 158

Im Listing »Sound.Ed.Zusatz« wurde eine Zeile vergessen. Fügen Sie deshalb folgende Zeile ein:
9990 REM

Profiprint, Ausgabe 11/85, Seite 74

Um die Fehlermeldung in Zeile 42 zu verhindern, ist die Zeile 1 wie folgt zu ergänzen:
1 CLR:POKE...

Neues vom EPROM-Markt, Ausgabe 12/85, Seite 11

Das in der rechten Spalte beschriebene CMOS-RAM wurde von Maas und Alaze entwickelt. Generalvertrieb für Deutschland hat Milan Rajĉić, Hammaner Str. 42, 4300 Essen 1, Tel. 0201/ 224141. Anfragen von Händlern sind an diese Adresse zu richten.

Prüfungsfragen, Ausgabe 9/85, Seite 57

Das in diesem Listing enthaltene Löschen von Sachgebieten funktioniert nicht. Die abgedruckte Fassung löscht statt des gewünschten Sachgebiets das benachbarte. Deshalb folgender Änderungsvorschlag:

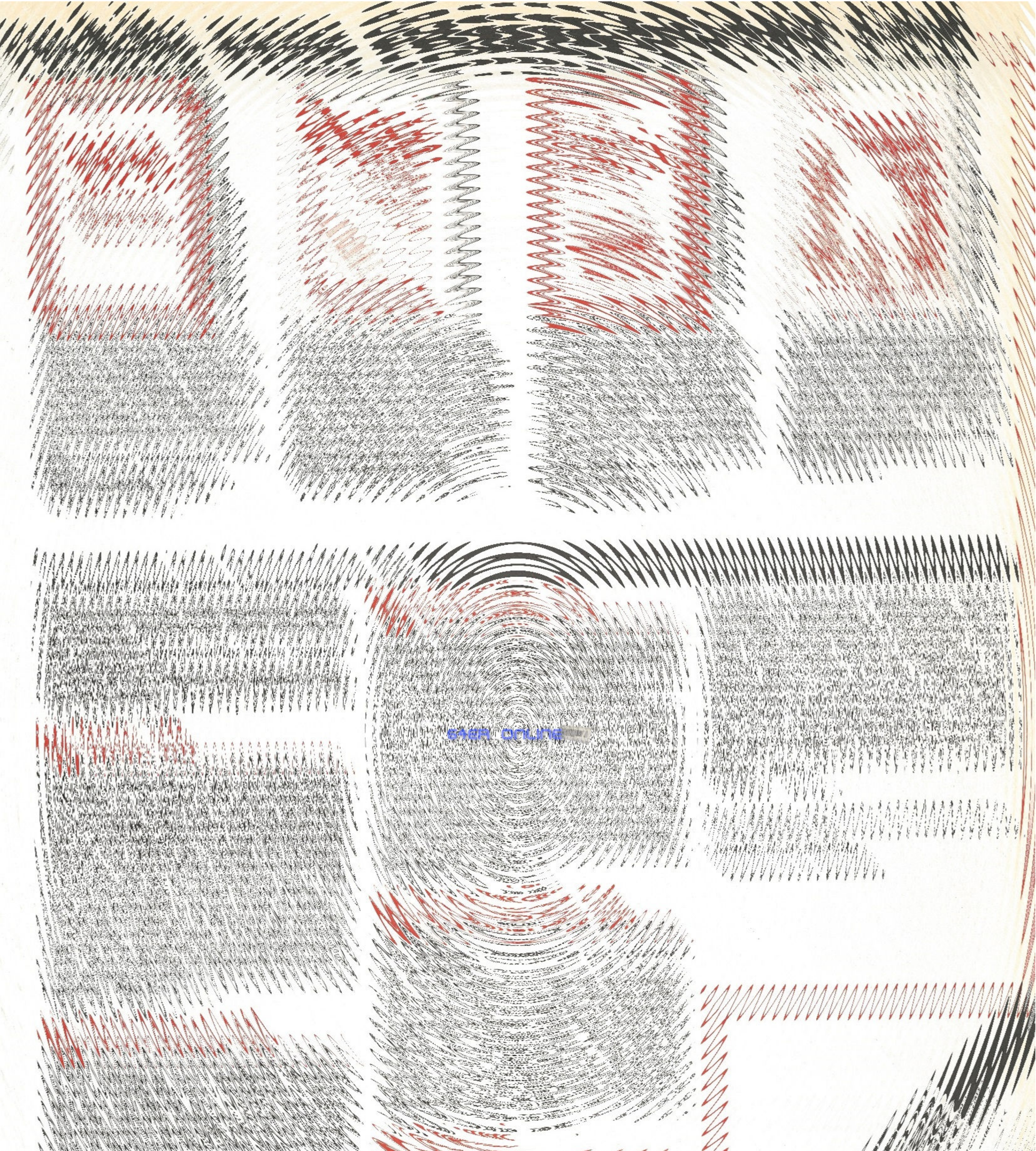
```
12766 OPEN 15,8,15,"S:"+SA$(A)+"/S":CLOSE 15
12767 OPEN 15,8,15,"S:?"+SA$(A):CLOSE 15
12768 OPEN 15,8,15, "S:??"+SA$(A):CLOSE 15
```

Die Zeilen 12830 bis 12832 sind ersatzlos zu streichen.

Hardcopy in doppelter Größe, Sonderheft 4, Seite 64

In Listing 1, Zeile 320 ist eine Zahl unleserlich abgedruckt. Bei ihr handelt es sich um die Zahl »41«; dann stimmt's.



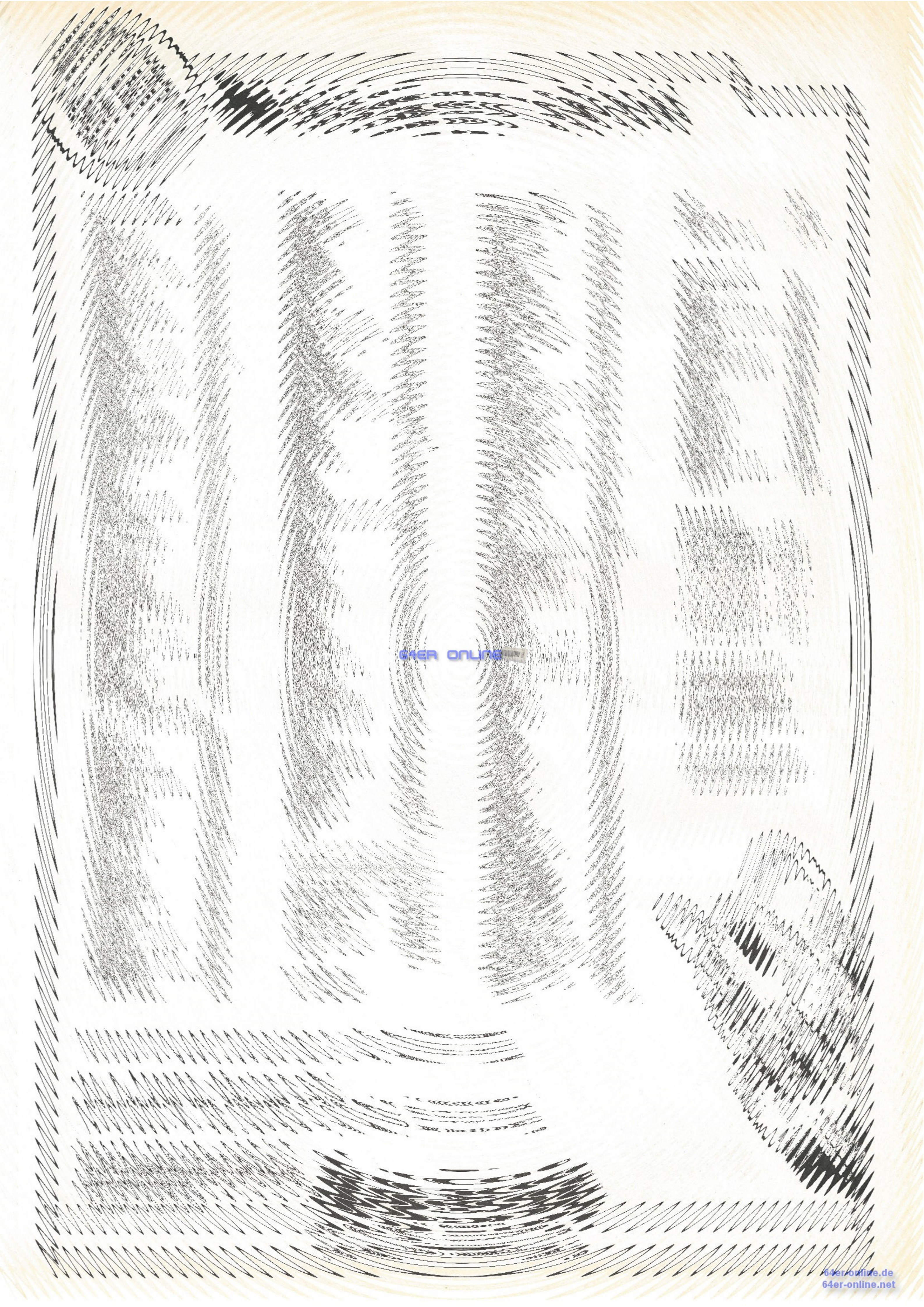


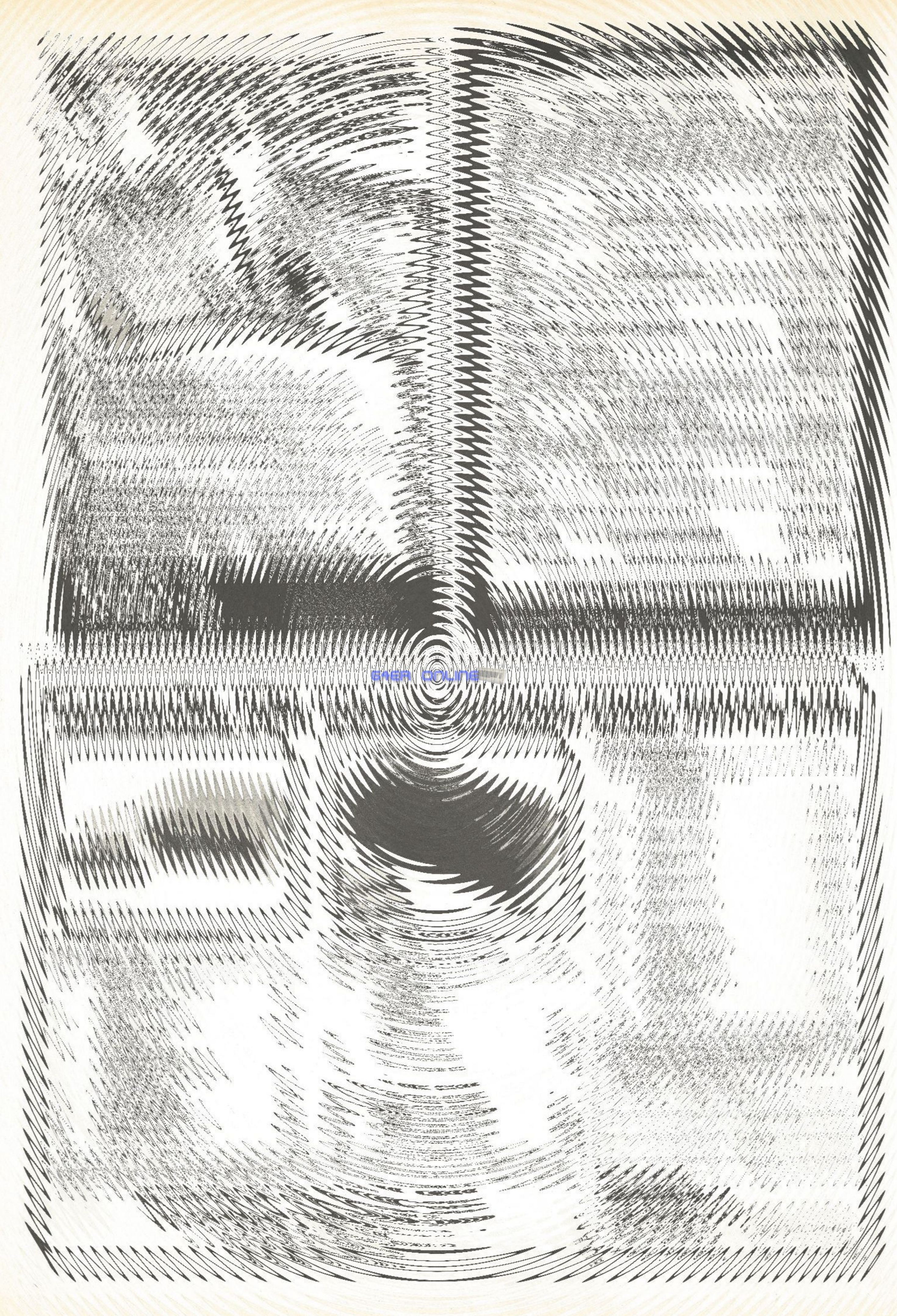
COMPUTER-MARKT

Wollen Sie einen gebrauchten Computer verkaufen oder erwerben? Suchen Sie Zubehör? Haben Sie Software anzubieten oder suchen Sie Programme oder Verbindungen? Der COMPUTER-MARKT von »Happy-Computer« bietet allen Computerfans die Gelegenheit, für nur 5,— DM eine private Kleinanzeige mit bis zu 5 Zeilen Text in der Rubrik Ihrer Wahl aufzugeben. Und so kommt Ihre private Kleinanzeige in den COMPUTER-MARKT der **März-Ausgabe** (erscheint am 14. Februar 86): Schicken Sie Ihren Anzeigentext bis zum 16. Januar 86 (Eingangsdatum beim Verlag) an »Happy-Computer«. Später eingehende Aufträge werden in der **April-Ausgabe** (erscheint am 14. März 86) veröffentlicht.

Am besten verwenden Sie dazu die vorbereitete Auftragskarte am Anfang des Heftes. Bitte beachten Sie: Ihr Anzeigentext darf maximal 5 Zeilen mit je 32 Buchstaben betragen. Überweisen Sie den Anzeigenpreis von DM 5,— auf das Postscheckkonto Nr. 14199-803 beim Postscheckamt mit dem Vermerk »Markt & Technik, Happy-Computer« oder schicken Sie uns DM 5,— als Scheck oder in Bargeld. Der Verlag behält sich die Veröffentlichung längerer Texte vor. Kleinanzeigen, die entsprechend gekennzeichnet sind, oder deren Text auf eine gewerbliche Tätigkeit schließen läßt, werden in der Rubrik »Gewerbliche Kleinanzeigen« zum Preis von DM 12,— je Zeile Text veröffentlicht.

Private Kleinanzeigen Private Kleinanzeigen Private Kleinanzeigen Private Kleinanzeigen





COMPUTER-MARKT

Private Kleinanzeigen Private Kleinanzeigen

Input 64 Kassetten 1/85-9/85, mit Heften (alles orig.): 65,— DM, 3 Stck. Arbeitsdisk.: SMON, Compiler, Disksorter, kopieren... 45,— DM (Syst. Wechsel), 04757/651

Suche defekten C-64 mit Fehlerbeschreibung, zahle bis zu 50 DM. Melden bei S. Pautzke, Osterlücken 17, 2398 Harrislee

Hallo C-64 Freaks!
Suche zuverl. Tauschpartner. Antwort bei Rückporto garantiert! M. Kraft, Neugasse 84, 6752 Winnweiler 1 (nur Disk)

Suche MIDI-Software für C64 mit DX-7, z.B. Editor, Sequenzer, Soundverwaltung und gute Sounds. Peter Kuvat, Ofterdingenstr. 36, 4300 Essen 14, Tel. 0201/530954

Österreich: Calc Result, Calc Result easy, diverse Data Becker Programme, MPS 802 Drucker, C64 Disketten und Hefte, Decathlon, Turbonibbler zu verkaufen, Tel. 4813093 (Wien)

Ich suche Tauschpartner (C64 Tape) für Spiele, Antwort bestimmt. Suche a. die beiden Bücher ★ Adventures u. Superspiele von Data Becker, Ans. Sandro, Friedrich Cosaarweg 39, 1 Berlin 20

Verkaufe: C64 + 1541 + 40-50 Hefte (alle 64'er) + 5 Bücher (4 von DB) + Zubehör, auch einzeln, Frithjof Dau, 3017 Pattensen 4, Hopfenberg 5, Tel. 05066/63906

Suche für den C-64 Wintergames. Kaufe oder tausche. Tel. 05127/1568, fragt nach Michael

Kaufe C64 + 1541 bis DM 500,—, 02162/53471

Zorro, Elite dtsch. hatte ich schon v. 6 Wochen. Suche Gleichgesinnte mit topakt. C-64 Softw. Auch Antlg. Listen an PLK 032028 C in 5090 Leverkusen 1. Ab die Post!

Verkaufe Wilke Assembler, Paint Pic, 64 Intern, Musikbuch, Maschinensprache für C-64 von Data Becker. Alles original, keine Kopie. Tel. 05174/1361

Suche defekten C64. Zahle bis 150 DM. Angebote bitte mit Fehlerbeschreibung, H. Sedlmeier, Marktstr. 21, 8396 Wegscheid, Tel. 08592/1329

Suche Tauschpartner aller Art. Wer tauscht elek. Schreibmaschine gegen 1541 oder MPS 801, 802, 803. Listen + Angeb. an Hubert Grasberger, Lerchenstr. 31, A-3380 Pöchlarn

■■■■■■■■■■■■■■■■■■■■
Suche zuverlässigen Tauschpart. nur Tape, antworte schnell, Listen an Wolfgang Mayer, Lehrstr. 36, 7210 Rottweil

Suche guterhaltene, funktionst. Floppy 1541, zahle 350 DM, Udo Lindner, Burgunderstr. 14, 8753 Obernburg, Tel. 06022/1744

Verk. C-64, Comm. Monitor 1701, Seikosha Drucker GP 500 A für 300 DM pro Teil. Sascha Lengen, Aachener Str. 43, 5102 Würselen. An C-64 Save Funktion kaputt.

Suche Floppy-1541! Preise nach Verhandlungsbasis! Angebote an: A. Drüppel, 6750 K'lautern, Erbsenberg 1

■■■ Assembler wechseln? ■■■
Quell-Texte retten durch Konvertierungsprogramme nur DM 15,—
■■■ Tel. 030/8613893 ■■■

C64 Suche Software (Disk) (z.B. Print Shop für MPS 802, neue Landschaftsdisk. zum Flight II, Kennedy Approach) M. Kummer, Linnefantstr. 12 a, 4650 Gelsenk.

Österreicher sucht Tauschpartner. Suche zuverl. Tauschpartner aus BRD, CH, ect. Habe Topgames für C-64. Antworte jedem, Liste an: Möbius, Sanddornstr. 5, 4600 Wels

Tausche Spiele und Anwenderprg.'s für C64 (nur Disk). Bitte melden bei Christian Goldschmidt (Witt)
Tel. (0231) 251266
Suche Newsroom und Bomb Jack

Suche Prgs. aller Art (bes. Sportspiele). Schickt Eure Listen mit Preisangaben an: St. Rennert, Seitenbach 20, 8601 Reckendorf
Bitte keine Wucherpreise!

★★★ Drucker VC-1526 ★★★
1 Jahr benutzt + Hardcopy-Programm für 520 DM. Mo-Fr ab 18 Uhr: 06851/2474
★★★★★★★★★★★★★★★★★★

Verkaufe und tausche C 64 Games. Habe: The Goonies, Street Hawk usw. Klaus Eichhorn, Friedelstr. 50, 1000 Berlin 44, Tel. 6241075

Tir. Na. Nog/Grogs Revenge je 20 DM Tapper, Indiana Jones, Hunchback 2 je 20 DM. Alle für C64 — Premierenbuch zum C128 25 DM, Jürgen Brechtel, 06233/28439

IEC Bus für VC 20 60 DM (NP 340 DM). Hiermit können Drucker, Plotter und Floppy der 2/3/4 + 8000er CBM Serien angeschlossen werden. J. Brechtel, 06233/28439

Verkaufe meine Anleitung Flugs. II (38 Seiten) + 4 Flugkarten DIN A3 für nur 10,— DM von Dieter Will, Ilsahl 13, 2350 Neumünster, Tel. 04321/31711

Verkaufe 64'er 4/84-10/85, Computer-Journal 1/82-3/84, HC, Run Data Welt (2x) zus. f. 150 DM, Roland Wartenberg, Lessingstr. 56, 6600 Saarbrücken 3

Betriebssystem für C64 m. Fast-Disk etc. und 4fach Umschaltplatine (Drehsch., absturzfrei) für zus. 50 DM oder einzeln à 27,—, Stefan Burkard, 0521/173038

Suche! Suche! Suche! Suche!
Wer kann ein Terminalprogramm nach Wünschen umprogrammieren? Es lohnt sich! Nach Klaus fragen, Telefon 02426/301

Suche alte o. defekte VC64 + Zubehör, 1541, 1001 o.2, CP80 X o. RX 80 günstig/kann bei Bauteilbesch. beh. sein, z.B. 6526 usw. ab 18.00, 0228/468016

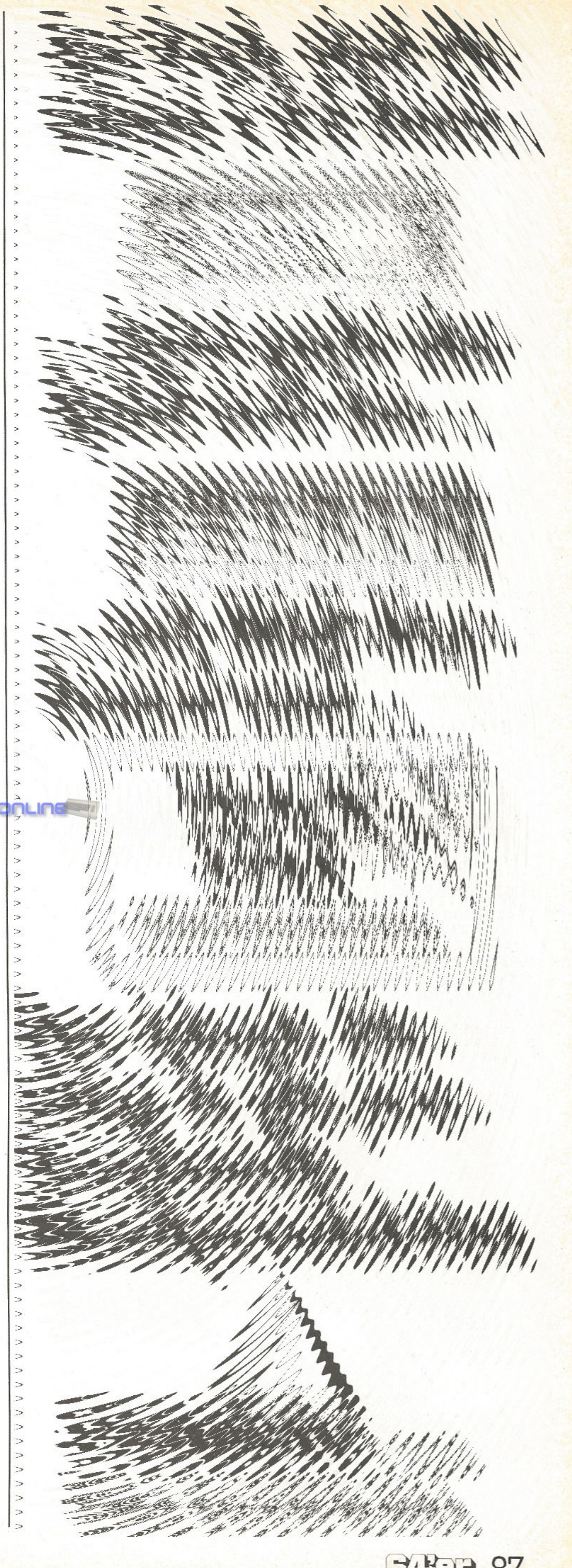
Speeddos für 100,— DM,
Tel. 06721/10097

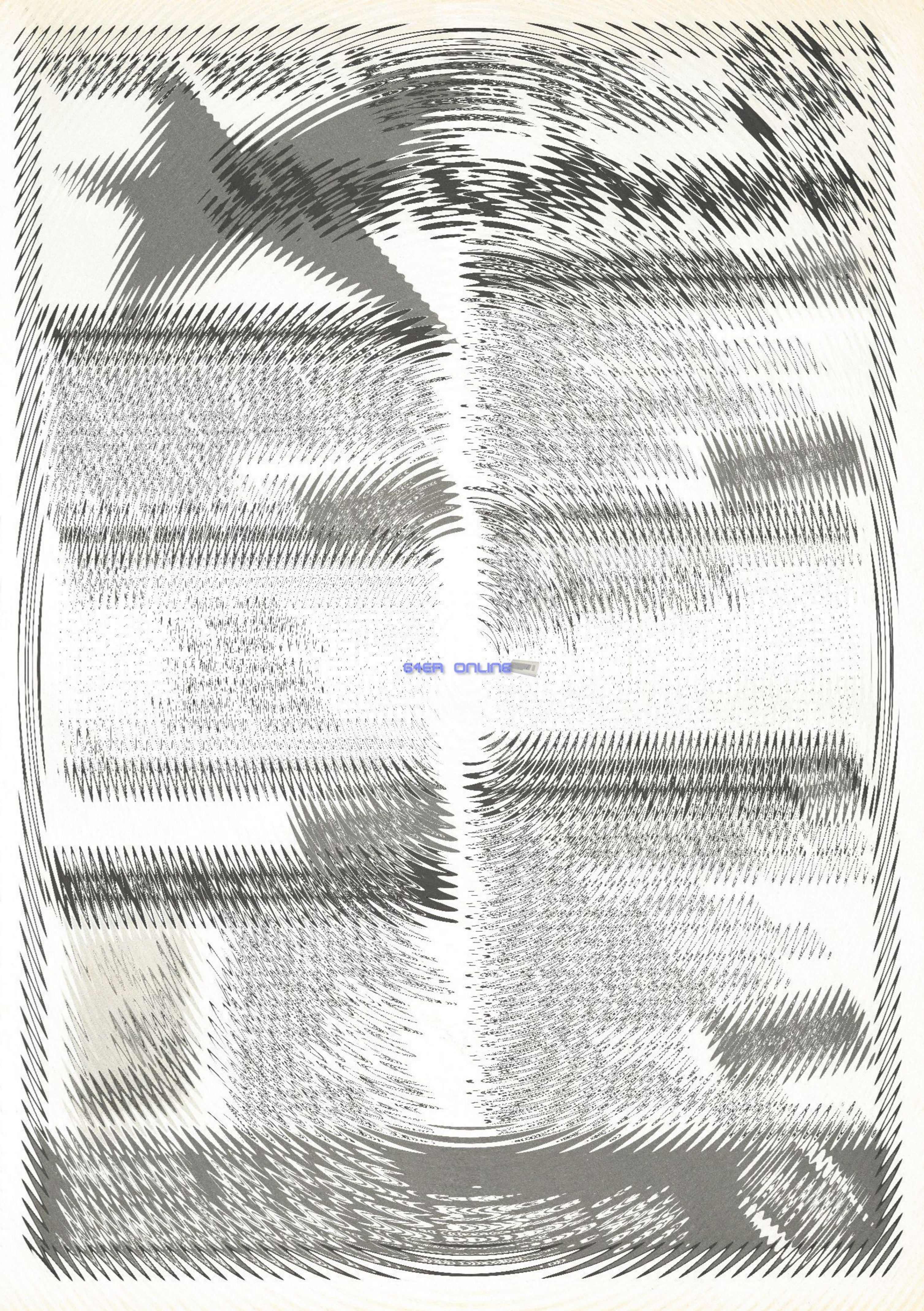
The Flight began on the 23/10/85 Glory to the Pirates, MAD, 4400 Münster — never Dying

Listing von Expertensystemen für ökologische Forschung (Jugend forscht) und Interessenten gesucht, Tel. 0521/21219

MPS 802/1526 druckt ä, ö, ü, sz durch unkomplizierten Hardware-Einbau. Auch komplett neue Zeichensätze möglich (Gotik, Western, Klassik, Comp.), VB 39,— DM. Tel. 05371/52809

Verkaufe
C64, Floppy 1541, Datasette und Drucker MPS 801 für 1250,— DM, Tel. 089/5804646




64ER ONLINE





64'er Extra

Das 64'er Extra bringt geballte Information über Ihren C 64 zum Heraustrennen und Sammeln.
In dieser fünften Ausgabe finden Sie den ersten Teil einer Übersicht über alle ROM-Routinen des C 64. Statt ziellos in ROM-Listings zu blättern, finden Sie hier im Klartext die Funktionsbeschreibung aller irgendwie nutzbaren Routinen.

BCOLD $A000
Basic-Kaltstartvektor ($E394). Unterzieht Basic der Prozedur NEW, gibt die Meldung BYTES FREE und READY aus. Dies beendet die Reset-Sequenz; siehe die Routinen bei $E394 und $FCE2.
BWARM $A002
Basic-Warmstartvektor ($E37B). Unterzieht Basic der Prozedur CLR, gibt READY aus. Dies steht am Ende der SHIFT-RESTORE-Sequenz; siehe die Routinen bei $E37B und $FE43.
$A004
Meldung des CBM-Basic.
$A00C
Tabelle der um 1 verminderten Adressen der Basic-Befehle (END, FOR, NEXT, etc.).
$A052
Tabelle der Adressen von Zahlen- und Stringfunktionen (SGN, INT, ABS, etc.).
$A080
Tabelle der um 1 verminderten Adressen der Routinen zur Bearbeitung der Basic-Operatoren (addieren, subtrahieren, dividieren, etc.); jeder Adresse folgt ein Byte, das die Priorität des Operators angibt.
$A09E
Basic-Schlüsselwörter als Zeichenketten im CBM-ASCII mit gesetztem Bit 7 des letzten Zeichens.
$A129
Tabelle verschiedener Schlüsselwörter, die keine Aktionsadressen haben, (TAB, TO, FN, etc.); Bit 7 des letzten Zeichens ist gesetzt.
$A140
Tabelle der Schlüsselzeichen der Opeatoren; außerdem AND, OR als Zeichenketten mit Bit 7 = '1' des letzten Zeichens.
$A14D
Tabelle der Schlüsselwörter der Funktionen (SGN, INT, ABS, etc.) mit gesetztem Bit 7 des letzten Zeichens.
$A19E
Tabelle der 28 Fehlermeldungen (TOO MANY FILES, FILE OPEN etc.) mit Bit 7 des letzten Zeichens gesetzt.
$A328
Tabelle der Zeiger zu den Fehlermeldungen
$A364
Tabelle der anderen Meldungen OK, ERROR IN, READY, BREAK.
FNDFOR $A38A
Prüft den Stapelspeicher auf den Eintrag für FOR. Aufgerufen von NEXT: wird FOR nicht gefunden, erscheint die Meldung ?NEXT WITHOUT FOR. Löscht auch, wenn von RETURN aufgerufen, einen FOR-Datenblock vom Stapel.
BLTU $A3B8
Schafft eine Lücke im Basic-Text zur Einfügung einer neuen Basic-Zeile. Prüft, ob genug Platz vorhanden, dann:
BLTUC $A3BF
Verschiebt den Block zwischen ($5F und ($5A)—1 hinauf zu einem neuen Blockende bei ($58)—1.
GETSTIK $A3FB
Prüft, ob der Stapelspeicher noch 2 * A Bytes faßt: wenn nicht, erscheint die Meldung ?OUT OF MEMORY.
REASON $A408
Prüft, ob der Zeiger in (A/Y) unterhalb FRETOP (aktuelle untere Grenze des Zeichenkettenbereichs) zeigt. Wenn ja, wird die Routine verlassen; wenn nein, erfolgt Garbage Collection: wenn danach immer noch nein, wird ?OUT OF MEMORY ausgegeben.
ERROR $A437
Gibt Fehlermeldung aus; X enthält die Fehlernummer (= die Hälfte der Versetzung innerhalb der Adressentabelle der Fehlermeldungen). Wird durch einen Vektor in ($0300) nach $A43A gelenkt. Dann werden die temporären Adressen für Tastatureingabe und Bildschirmausgabe gesetzt, der Steckpointer zurückgesetzt und, falls im Programm-Modus, die Fehlermeldung zusammen mit der Zeilennummer ausgegeben. Dann:
READY $A474
Startet Basic neu: Gibt READY aus, schaltet auf Direktmodus, dann:
MAIN $A480
Nimmt eine Zeile in den Eingabepuffer herein und fügt ein abschließendes Nullbyte an. Prüft, ob Programmzeile oder Befehl im Direktmodus: führt letzteren sofort aus. (MAIN wird über einen Vektor in ($0302) zu $A483 weitergeleitet.)
MAIN1 $A49C
Ist es eine Programmzeile, wird sie verschlüsselt.
INSLIN $A4A4
Ersetzt die alte Zeile durch die neue, falls die Zeilennummer bereits vorhanden; fügt ansonsten die neue Zeile ein. Die Zeilennummer befindet sich beim Eintritt in $14, $15; die Länge +4 steht in Y. Ist das erste Byte im Puffer gleich 0, handelt es sich um eine leere Zeile: die alte Zeile wird gelöscht.
FINI $A52A
Führt nach dem Einfügen einer neuen Zeile RUNC (dadurch gehen beim Editieren die Variablen verloren und anschließendes CONT ist nicht mehr möglich) und LNKPRG aus; springt anschließend zu MAIN.
LNKPRG $A533
Verkettet die Bindezeiger im Basic-Programm; verwendet dabei Nullbytes als Markierungen für das Zeilenende.
INLIN $A560
Nimmt eine Bildschirmzeile in den Basic-Puffer ab $0200 herein und fügt ein abschließendes Nullbyte an.
RDCHR
Setzt einzelnes Zeichen in A.
CRUNCH $A579
Verschlüsselt die Schlüsselwörter im Eingabepuffer. Wird über einen Vektor in ($0304) gewöhnlich nach $A57C gelenkt.
FNDLIN $A613
Sucht den Basic-Text von Anfang an nach der Zeilennummer in $14, $15 ab. Bei gefundener Zeilennummer wird das Übertragsflag gesetzt; ($5F) zeigt auf die Bindeadresse.
FNDLNC $A617
Sucht Basic-Text ab A (niederwertiges Byte) und Y (höherwertiges Byte) nach der Zeilennummer in $14, $15 ab.
SCRATH $A642
***NEW springt hierher: prüft die Syntax, dann:
SCRTCH $A644
Setzt die ersten 2 Byte des Textes (= der erste Bindezeiger) auf 0; lädt den Zeiger für den Anfang der Variablen $2D,$2E mit dem Anfang von Basic+2, dann:
RUNC $A659
Setzt mittels STXPT den Zeiger innerhalb GETCHR auf den Anfang des Basic-Texts, dann:
CLEAR $A65E
***CLR: Löscht die Variablen durch Zurücksetzen der Zeiger zum Ende des Variablenbereichs auf den Wert des Zeigers für das Programmende; eventuelle Zeiger von Zeichenkettenvariablen werden ebenfalls zurückgesetzt. Bricht I/O-Aktivitäten ab und setzt den Stackpointer zurück.
STXPT $A68E
Setzt den Zeiger innerhalb der Routine GETCHR auf den Anfang des Basic-Texts zurück. (das heißt, lädt $7A,$7B mit $2B,2C-1).
LIST $A69C
*** EINSPRUNG für den LIST-Befehl.
LIST1 $A6C9
Listet eine Zeile Basic: zuerst die Zeilennummer, dann den Text.
QPLOP $A717
Bearbeitet das zu listende Zeichen: ist es ein gewöhnliches oder ein Steuerzeichen in Anführungszeichen, wird es ausgegeben; entschlüsselt und druckt Schlüsselzeichen über einen Vektor in ($0306) nach $A71A gelenkt.
FOR $A742
*** Bearbeitung von FOR. Schiebt 18 Byte auf den Stapelspeicher. 1) Zeiger auf die nächstfolgende Anweisung, 2) momentane Zeilennummer, 3) oberer Schleifenwert, 4) Schrittweite (Standardwert = 1), 5) Name der Schleifenvariablen und 6) das Schlüsselzeichen für FOR.
NEWSTT $A7AE
Führt Basic aus: Prüft auf die STOP-Taste und auf das Nullbyte für das Zeilenende oder auf einen Doppelpunkt.
CKEOL $A7C4
Hält an, wenn Textende erreicht und setzt andernfalls den Zeiger innerhalb CHRGET auf den Anfang der nächsten Zeile.
GONE $A7E1
Bearbeitet die Basic-Anweisung in der momentanen Zeile. Über einen Vektor in ($0308) nach A7E4 gelenkt; kehrt in einer Schleife zurück zu NEWSTT.
GONE3 $A7ED
Führt ein Basic-Schlüsselwort aus. Bezieht die Adresse zum Starten der Routine aus der Tabelle bei $A00C. Nimmt "LET" an, wenn das erste Byte der Anweisung kein Schlüsselwort ist. Legt Adresse auf den Stapelspeicher, so daß der RTS-Befehl von CHRGET dorthin springt.
RESTOR $A81D
***RESTORE: Setzt den Datenzeiger bei $41, $42 auf den Anfang des Basic-Textes.
STOP $A82C
***STOP: auch END sowie Programmunterbrechung. Speichert Information für CONT (Zeiger in den Basic-Text, Zeilennummer). STOP gibt BREAK IN xxx aus, während END dies überspringt und stattdessen READY druckt. Die STOP-Taste aktiviert STOP; Ein Nullbyte für das Programmende ruft END auf.
CONT $A857
*** CONT: Setzt den Programmlauf fort. Setzt hierzu die aktuelle Zeilennummer ($39, $3A) und den Zeiger innerhalb CHRGET auf die von STOP gespeicherten Werte. ?CANT'T CONTINUE ERROR erscheint, wenn das höherwertige Byte des Zeigers bei einem Syntax-Fehler auf »0« gesetzt wurde.
RUN $A871
*** RUN: Löscht alle Variablen (CLR), setzt den Stapelspeicher zurück, setzt CHRGET auf den Anfang von Basic und beginnt mit der Ausführung des Programms. Auf RUN xxx folgt CLR aller Variablen, das Zurücksetzen des Stapelspeichers und schließlich GOTO xxx.
GOSUB $A883
*** GOSUB: Schiebt 5 Byte auf den Stapelspeicher: 1) den Zeiger innerhalb CHRGET, 2) die aktuelle Zeilennummer, 3) das Schlüsselzeichen für GOSUB; anschließend wird GOTO aufgerufen.
GOTO $A8A0
*** GOTO: Holt sich die auf die GOTO-Anweisung folgende Zeilennummer und sucht den Basic-Text nach dieser Zeile ab. Ist das höherwertige Byte der Bestimmungszeile größer als das der aktuellen Zeilennummer, erfolgt die Suche ab der momentanen Zeile aufwärts, um die Suchzeit abzukürzen; sonst beginnt die Suche von Anfang an. Setzt Zeiger auf die gefundene Zeile in CHRGET ein.
RETURN $A8D2
*** RETURN: Der Stapelspeicher wird bis zum GOSUB-Schlüsselzeichen restorniert (?RETURN WITHOUT GOSUB, wenn keines gefunden). Anschließend werden die Nummer und der Zeiger der aufrufenden Zeile wiederhergestellt und die Programmausführung fortgesetzt.
DATA $A8F8
*** DATA: Die Routine veranlaßt CHRGET, die DATA-Anweisung bis zum abschließenden Nullbyte oder einem Doppelpunkt zu überspringen.
DATAN $A906
Sucht nach der Endemarkierung der Anweisung; beim Aussprung enthält Y den Unterschied zwischen Zeilenende und dem Zeiger der Routine CHRGET.
REMN $A909
Sucht nach dem Ende einer Basic-Zeile.

IF $A928
* * * IF: Wertet den Ausdruck aus; ist das Resultat »Falsch« (das heißt gleich 0), wird der THEN- oder GOTO-Teil mittels REM übergangen.

REM $A93B
* * * REM: Sucht nach dem Ende der Zeile und bringt den Zeiger in CHRGET auf den neuen Stand. Dadurch wird der Inhalt der REM-Zeile übergangen.

DOCOND $A940
Setzt IF fort: Ist der Ausdruck wahr, wird der nächste Befehl, oder, wenn eine Zahl folgt, GOTO ausgeführt.

ONGOTO $A94B
* * * ON: Berechnet den Ausdruck, prüft auf Schlüsselzeichen für GOTO oder GOSUB, geht die Liste der Zeilennummern nach der angegebenen Zeilennummer durch, überspringt dabei Kommata, und führt schließlich GOTO oder GOSUB dorthin aus.

LINGET $A96B
Liest aus dem Basic-Text eine Ganzzahl (gewöhnlich eine Zeilennummer) in $14, $15 ein; die Zahl muß im Bereich 0...63999 liegen.

LET $A9A5
* * * LET: Sucht Zielvariable in der Variablenliste (oder erzeugt sie, falls nicht vorhanden), prüft auf das Schlüsselzeichen für "=", berechnet den Ausdruck und setzt das Ergebnis oder den Zeichenkettendescriptor in die Variablenliste.

PUTINT $A9C4
Rundet den FAC1 und setzt ihn als Ganzzahl in die Variablenliste an die aktuelle Variablenposition, auf die ($49) zeigt.

PTFLPT $A9D6
Setzt FAC1 in die Variablenliste an die durch ($49) angegebene Stelle.

PUTTIM $A9E3
Ordnet die Systemvariable TI$ zu.

ASCADD $AA27
Addiert ASCII-Ziffer zu FAC1.

GETSPT $AA2C
LET für Zeichenketten — setzt den Zeichenketten-Deskriptor, auf den (FAC1 + 3) zeigt, in die Variablenliste an die Stelle ($49).

PRINTN $AA80
PRINT: Ruft CMD auf, löscht dann die I/O-Kanäle und setzt die Standard-Geräteadressen wieder ein.

CMD $AA86
* * * CMD: Setzt mittels der Kernel-Routine CHKOUT das CMD-Ausgabegerät aus der Dateitabelle und ruft PRINT auf.

STRDON $AA9A
Teil der PRINT-Routine: Gibt Zeichenkette aus und fährt mit der Abwicklung von PRINT fort.

PRINT $AAA0
* * * PRINT: Identifiziert PRINT-Parameter (TAB, SPC, Komma, Strichpunkt usw.) und berechnet Ausdrücke.

VAROP $AAB8
Gibt Variable aus; Zahlen werden zunächst in eine Zeichenkette umgewandelt.

CRDO $AAD7
Gibt eine Zeilenschaltung aus, der ein Zeilenvorschub folgt, wenn die Kanaladresse 128 ist.

STROUT $AB1E
Druckt die Zeichenkette, auf die (A/Y) zeigt, bis ein Nullbyte oder Anführungszeichen angetroffen wird.

STRPRT $AB21
Druckt Zeichenkette: (FAC1+3) zeigt auf den Zeichenketten-Deskriptor.

OUTSTR $AB24
Zeichenkettenausgabe: ($22) zeigt auf die Zeichenkette; die Länge ist in A festgehalten.

OUTSPC $AB3B
Ausgabe eines Cursorschritts nach rechts (oder eines Leerzeichens, wenn die Ausgabe nicht zum Bildschirm erfolgt).

PRTSPC $AB3F
Gibt Leerzeichen aus.

OUTSKP $AB42
Gibt Cursorschritt rechts aus.

OUTQST $AB45
Gibt Fragezeichen vor Fehlermeldung aus.

OUTDO $AB47
Gibt das Zeichen in A aus.

TRMNOK $AB4D
Gibt die entsprechenden Fehlermeldungen für GET, READ und INPUT aus.

GET $AB7B
* * * GET: Prüft auf Direktmodus (nicht zulässig) und holt ein Zeichen von der Tastatur oder Datei.

INPUTN $ABA5
* * * INPUT: Holt die Dateinummer, schaltet das Gerät ein, ruft INPUT auf und schaltet dann das Gerät ab.

INPUT $ABBF
* * * INPUT: Gibt den Hinweistext des Benutzers aus, falls vorhanden, dann:

QINLIN $ABF9
Bringt ein »?« auf den Bildschirm und nimmt die durch eine Zeilenschaltung begrenzte Zeile Text in den Eingabepuffer herein.

READ $AC06
* * * READ: GET und INPUT teilen sich diese Routine, werden aber durch ein Flag in $11 unterschieden.

INPCON $AC0D
Einsprung für INPUT: Setzt das Flag und ruft READ auf; (X/Y) zeigt auf den Puffer.

INPC01 $AC0F
Einsprung für GET: Setzt das Flag und ruft READ auf; (X/Y) zeigt auf den Puffer.

DATLOP $ACB8
Geht Text durch und liest DATA-Anweisungen

VAREND $ACDF
Prüft auf eine »0« am Ende des Eingabepuffers; gibt die Meldung? EXTRA IGNORED aus, wenn nicht gefunden.

EXINT $ACFC
Die Meldungen? EXTRA IGNORED und ?REDO FROM START beide gefolgt von CRLF (Zeilenschaltung und Zeilenvorschub).

NEXT$AD1E
* * * NEXT: Prüft auf das Schlüsselzeichen für FOR und die passende Variable auf dem Stapelspeicher; gibt ?NEXT WITHOUT FOR aus, wenn nicht gefunden; berechnet nächsten Wert: wenn dieser noch gültig, werden die aktuelle Zeilennummer und der Zeiger in CHRGET zurückgesetzt und die Schleife fortgesetzt.

FRMNUM $AD8A
Berechnet einen numerischen Ausdruck von Basic aus durch Aufruf von FRMEVL, dann:

CHKNUM $AD8D
Prüft durch Testen der Flag bei $0D, ob FRMEVL eine Zahl geliefert hat: Meldung ?TYPE MISMATCH ERROR, wenn nicht.

CHKSTR $AD8F
Prüft durch Testen der Flag bei $0D, ob FRMEVL eine Zeichenkette geliefert hat: Meldung ?TYPE MISMATCH ERROR, wenn nicht.

FRMEVL $AD9E
Wertet jeglichen Basic-Formelausdruck aus und meldet alle Syntaxfehler; setzt $0D (VALTYP) im Falle einer Zahl auf $00, bei einer Zeichenkette auf $FF. Setzt bei einer Fließkommazahl $0E (INTFLG) auf $00 und legt den Wertin FAC1 ab. War der Variablentyp ganzzahlig, wird INTFLG auf $80 gesetzt, das Ergebnis jedoch im Fließkommaformat in FAC1 belassen. Komplizierte Ausdrücke können eine Vereinfachung erfordern, um Platz im Stapelspeicher zu gewinnen und den Fehler ?OUT OF MEMORY zu vermeiden.

EVAL $AE83
Wertet einzelnes Glied in einem Ausdruck aus; sucht nach Ziffernketten in ASCII, Variablen, π, NOT, arithmetischen Funktionen und so weiter.

PIVAL $AEA8
Wert für π im 5-Byte-Fließkommaformat.

PARCHK $AEF1
Wertet Klammerausdrücke innerhalb des Textes aus.

CHKCLS $AEF7
Prüft, ob CHRGET auf ")" zeigt —?SYNTAX ERROR, wenn nicht.

CHKOPN $AEFA
Prüft, ob CHRGET auf "(" zeigt —?SYNTAX ERROR, wenn nicht.

CHKCOM $AEFD
Prüft, ob CHRGET auf " , " zeigt —?SYNTAX ERROR, wenn nicht.

SYNCHR $AEFF
Prüft, ob CHRGET auf ein Byte gleich dem in A zeigt: wenn ja, wird die Routine mit dem nächsten Byte in A verlassen, sonst ?SYNTAX ERROR.

SYNERR $AF08
Gibt ?SYNTAX ERROR aus und kehrt zum READY-Status zurück.

DOMIN $AF0D
Berechnet NOT.

TSTROM $AF14
Setzt das Übertragsflag auf 1, wenn (FAC1+3) auf ROM zeigt, was auf eine der reservierten Variablen TI$, TI, ST hinweist.

ISVAR $AF28
Sucht die Variablenliste nach der in $45, $46 genannten Variablen ab; beim Verlassen der Routine enthält FAC1 einen numerischen Wert im FLPT-Format (sowohl bei Ganzzahl- wie bei Fließkommavariablen); im Falle einer Stringvariablen zeigt (FAC1+3) auf den Zeichenketten-Descriptor.

TISASC $AF48
Liest den Taktzähler und bildet eine Zeichenkette, die TI$ enthält.

ISFUN $AFA7
Stellt Funktionstyp fest und wertet ihn aus.

OROP $AFE6
* * * OR-Funktion: Setzt Flag und führt mit zwei Ganzzahlen aus je 2 Bytes in FAC1 und FAC2 logisches ODER (OR) aus.

ANDOP $AFE9
* * * AND-Funktion: Diese Operationen werden durch eine Routine ausgeführt; ein Flag (in Y) enthält $FF für OR, $00 für AND. Wandel FLPT in Ganzzahl um (Fehlermeldung, wenn außerhalb des zulässigen Bereichs); das Ergebnis bleibt im FLPT-Format in FAC1.

DOREL $B016
* * * Vergleiche: < = > prüfen die Variablentypen, dann:

NUMREL $B01B
Führt mittels FCOMP bei $DC5B numerische Vergleiche aus, oder:

STRREL $B02E
Führt Zeichenkettenvergleich aus: Aussprung mit X=0 bei gleichen Zeichenketten mit X=1, wenn die erste > die zweite und X=255, wenn die zweite > die erste.

DIM $B081
* * * DIM: Stellt mittels der Routine PRTGET alle Feldelemente auf.

PTRGET $B08B
Prüft einen Variablennamen in Basic auf Zulässigkeit; das erste Zeichen muß ein Buchstabe sein, das zweite ein Buchstabe oder eine Zahl; was danach kommt, bleibt unbeachtet. Setzt VALTYP ($0D) auf $FF; wenn ein »$« angetroffen wird, sonst auf $00; setzt INTFLG ($0E) auf $80, wenn ein »%« gefunden wird. Der Name wird in VARNAM ($45, $46) zwischengespeichert. Dabei sind die höchsten Bits so gesetzt, daß sie über den Variablentyp Auskunft geben, wie in Kapitel 5 beschrieben.

ORDVAR $B0E7
Sucht die Variablenliste nach dem in VARNAM ($45, $46) abgelegten Namen ab und läßt VARPNT ($47) darauf zeigen. Definiert neue Variable, wenn Name nicht gefunden.

ISLETC $B113
Setzt das Übertragungsflag, wenn der Akkumulator einen der Buchstaben A...Z enthält.

NOTFNS $B11D
Stellt eine neue einfache Variable (keine Feldvariable) in der Variablenliste unmittelbar vor den Feldern auf. Der Name befindet sich in VARNAM ($45, $46). Eventuelle Felder müssen um 7 Byte nach oben rücken, um Platz für die neue Variable zu schaffen. Beim Verlassen zeigt ($5F) auf die neu eingerichtete Variable.

FMAPTR $B194
Berechnet Zeigerwert in $5F,$60; wird bei der Zuweisung von Platz für Felder gebraucht.

N32768 $B1A5
Enthält — 32768 als 5-Byte-Fließkommazahl.

FACINX $B1AA
Verwandelt FAC1 in eine Ganzzahl (—32768 bis +32767) und speichert sie in A/Y.

INTIDX $B1B2
Holt einen positiven Ganzzahlausdruck vom nächsten Teil des Basic-Texts und wertet ihn aus; liegt das Resultat im Bereich 0...32767, wird es in (FAC1+4, FAC1+3) abgelegt.

AYINT $B1BF
Wandelt FAC1 in Ganzzahl des Bereichs 0...32767 um; Ergebnis in FAC+3.

ISARY $B1D1
Holt Feldparameter aus dem Basic-Text: Zahl der Dimensionen, Zahl der Elemente. Legt sie auf den Stapelspeicher.

FNDARY $B218
Sucht nach dem Feldnamen in VARNAM ($45,$46) und nach den anderen Einzelheiten auf dem Stapel.

BSERR $B245
BAD SUBSCRIPT ERROR (falscher Index). BS+3 = ILLEGAL QUANTITY ERROR (nicht zulässige Größe).

NOTFDD $B261
Ist das Feld nicht vorhanden, nimmt diese Routine die Angaben auf dem Stapelspeicher und definiert das Feld mit der Dimension 10.

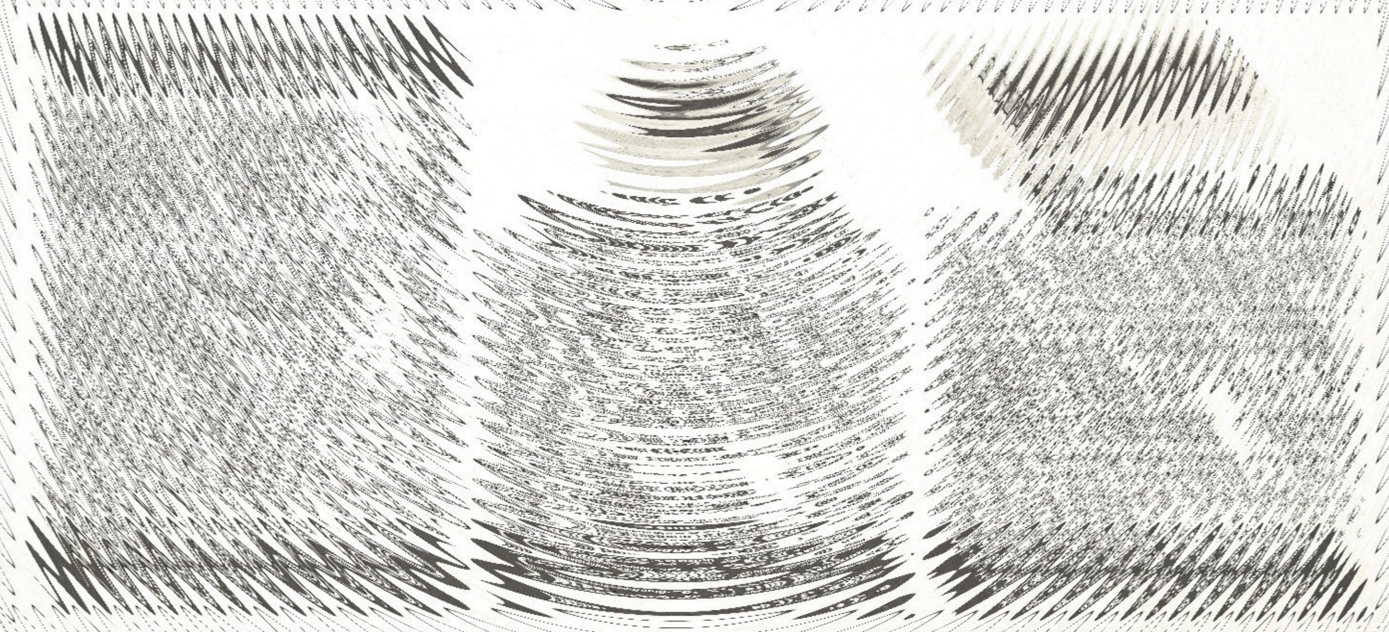
INPLN2 $B30E
Macht angegebenes Feldelement ausfindig und läßt VARPTN ($47) darauf zeigen.

UMULT $B34C
Berechnet den Abstand des angegebenen Feldelements von der Feldadresse, auf die (VARPTN) ($47) zeigt; setzt das Ergebnis in X/Y.

FRE $B37D
* * * FRE: Führt Garbage Collection aus und läßt (Y/A) auf die unterste (Zeichenkette minus Zeiger auf das Ende der Felder) zeigen; plaziert das Ergebnis in FAC1 durch Aufruf von:

GIVAYF $B391
Wandelt die 2-Byte-Ganzzahl in Y/A (Bereich —32768 bis +32767) in FLPT um und setzt sie in FAC1.

Fortsetzung im nächsten Extra

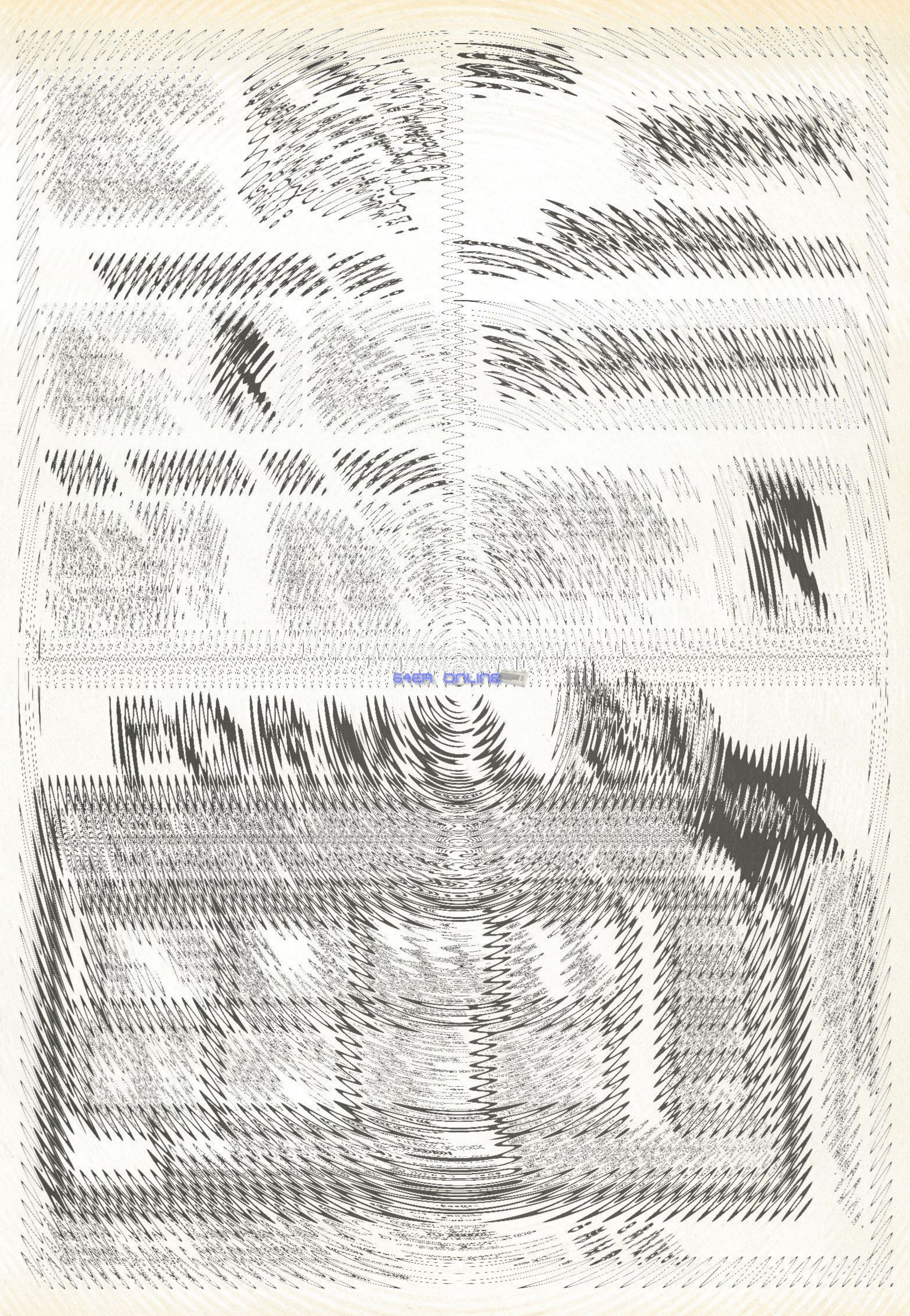


COMPUTER-MARKT

Private Kleinanzeigen Private Kleinanzeigen







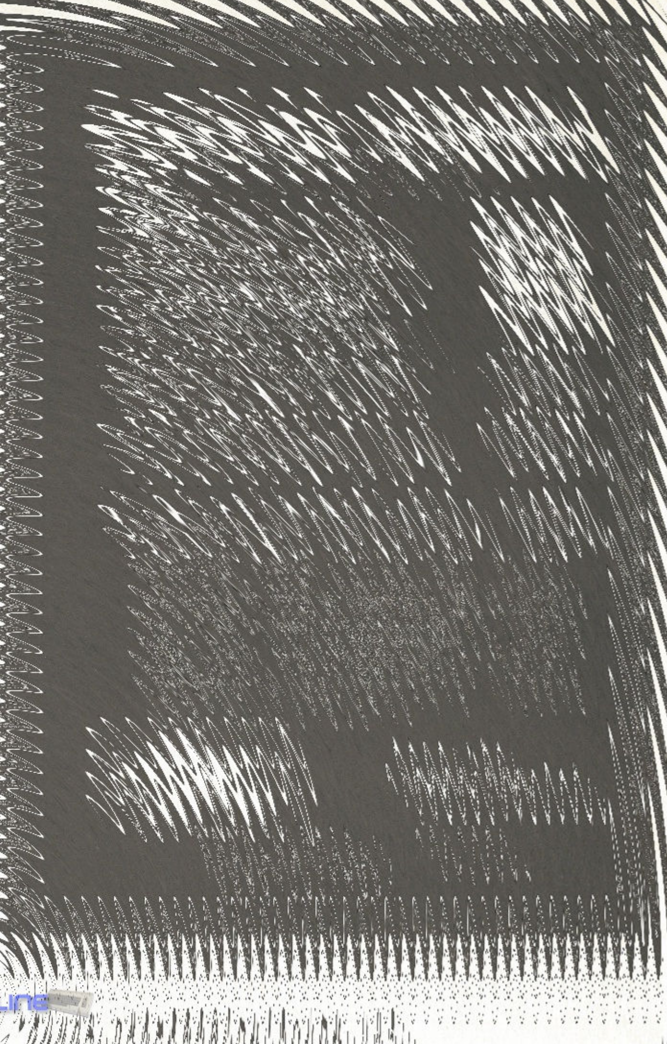






COMPUTER-MARKT

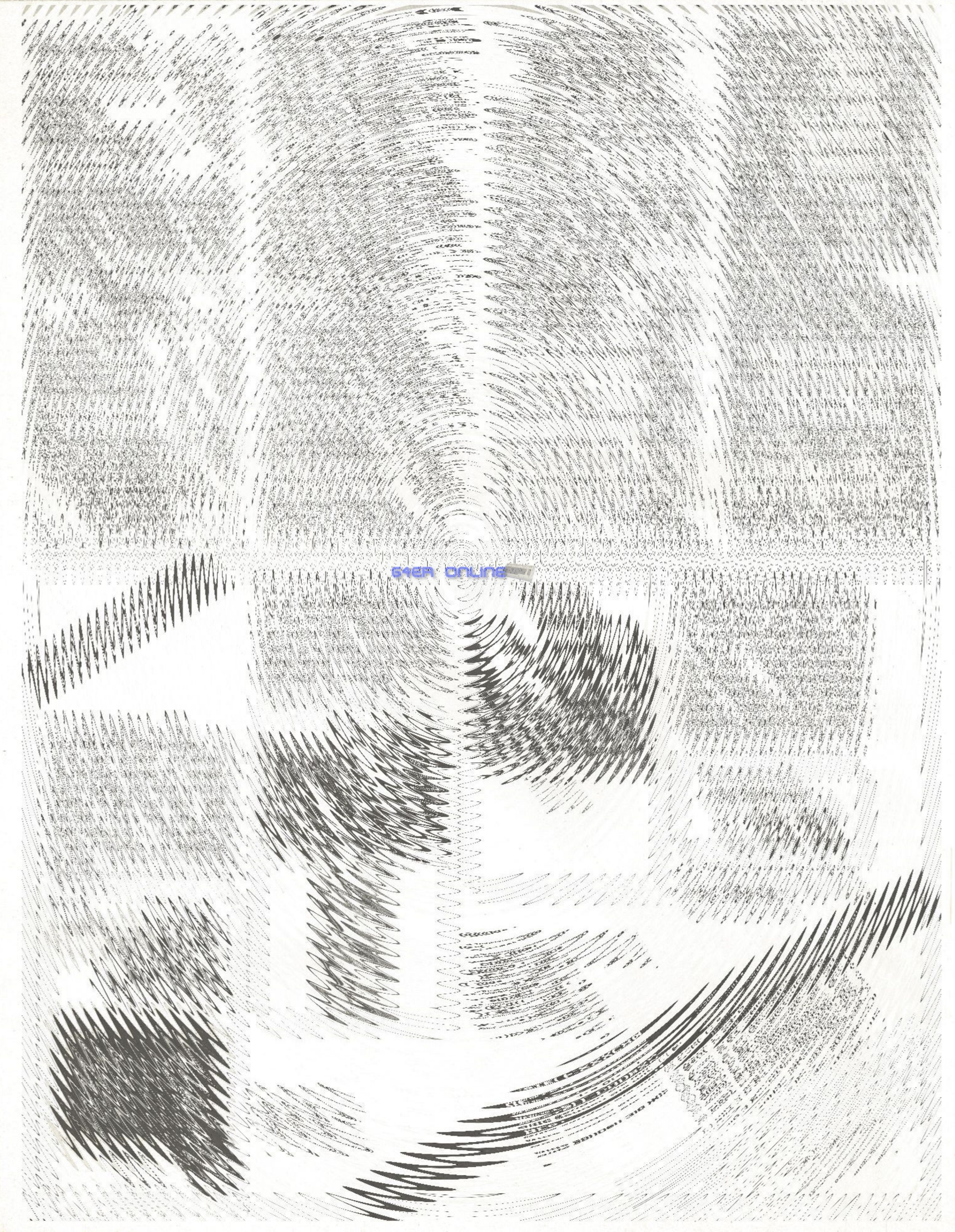
Private Kleinanzeigen Private Kleinanzeigen



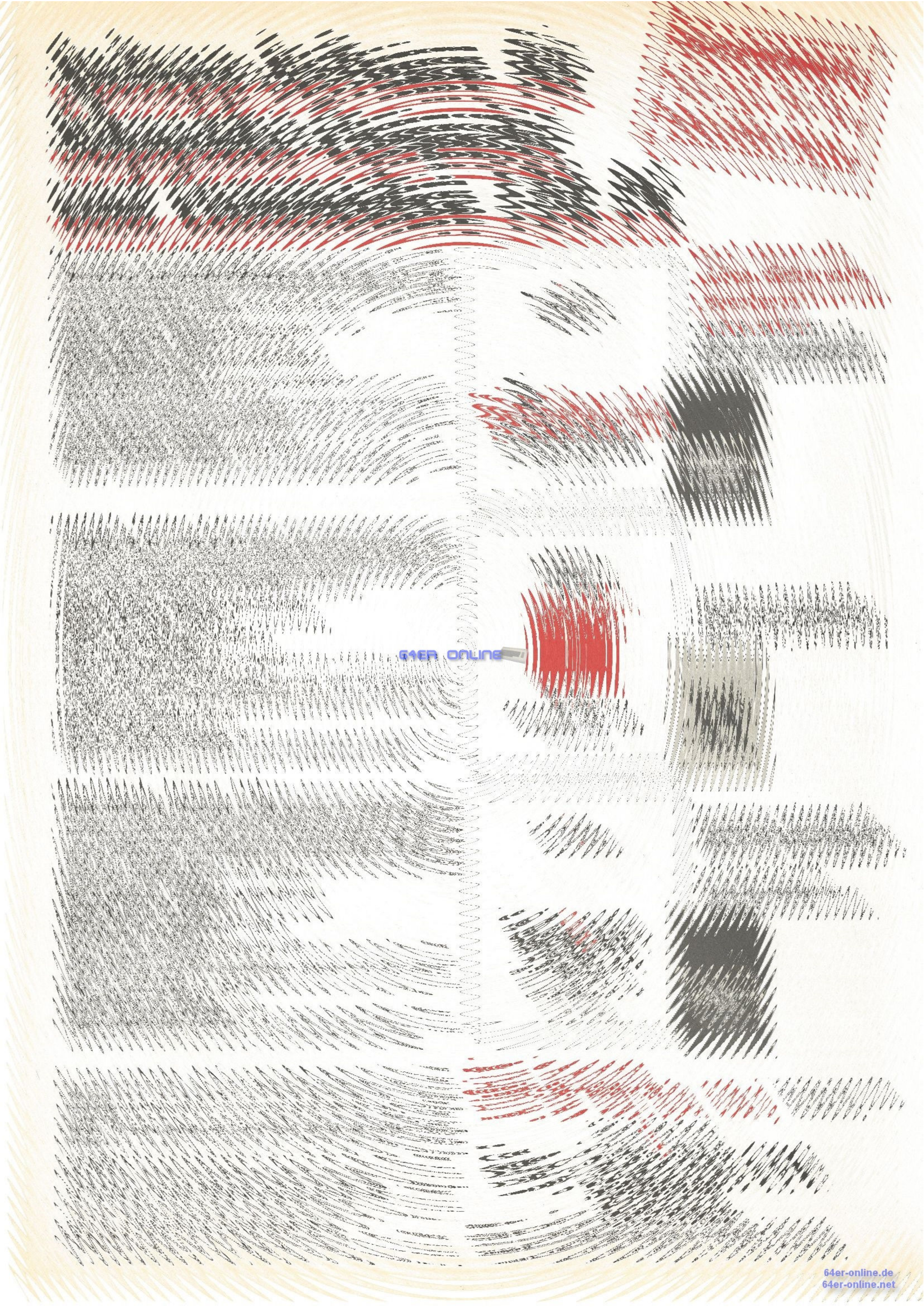




 Gewerbliche Kleinanzeigen







 Gewerbliche Kleinanzeigen



Gewerbliche Kleinanzeigen

Gewerbliche Kleinanzeigen





Der Commodore 64

Es fällt schwer, über das aus dem Amerikanischen übersetzte Buch von Tim Onosko ein einheitliches Urteil abzugeben, zu unterschiedlich sind die beiden Hälften. Während der erste Teil für den Anfänger, der noch keine Ahnung hat, geschrieben ist, findet im zweiten Teil auch der erfahrene C 64-Besitzer lesenswerte Abschnitte.

In typisch amerikanischer Frische wird der Leser im ersten Teil ganz ausführlich mit dem C 64 und der Basic-Programmierung vertraut gemacht. Einfach nachzuvollziehende Beispiele zeigen dem Anfänger, wie er seinen C 64 nutzbringend einsetzen kann. Er erfährt, daß der »Einsatz des Computers das Leben erleichtern, aber nicht komplizieren« soll. Deutliche Kritik muß dem Abschnitt über käufliche Textverarbeitungsprogramme ausgesprochen werden. Es zeugt bestimmt von Gedankenlosigkeit der Übersetzer, wenn sie einfach die amerikanischen Ausführungen über hier recht unbekannte Programme übernehmen. Eine der wichtigsten Fragen für einen deutschen Benutzer eines Textverarbeitungsprogrammes ist doch bestimmt die Frage, ob das Programm die Umlaute und ß ausgeben kann. Der Leser wird hierüber leider im Unklaren gelassen, statt dessen erfährt er die amerikanischen Lieferanschriften. Über Textomat und Vizawrite, die für den C 64 im deutschen Sprachraum wahrscheinlich am meisten verbreiteten Programme, ist kein Wort zu lesen.

Wesentlich interessanter ist der Anhang, der gut ein Drittel des Buches ausmacht. Von Jim Butterfield wird der Commodore 64 erforscht. Ganze Kapitel, die auch viele wertvolle Tabellen beinhalten, sind der Grafik und der Musik gewidmet. Hier wird wirklich gründlich, dabei aber auch verständlich informiert. (D. Hein/ev)

Info: Tim Onosko, Der Commodore 64 für Hobby, Schule und Beruf, Carl Hanser Verlag, 342 Seiten, ISBN 3-446-14073-5, Preis 48 Mark

Das Musikbuch zum Commodore 64

Data Becker hat sich mit dem schon seit einiger Zeit erhältlichen, aber immer noch aktuellen Musikbuch einer wichtigen Fähigkeit des Commodore 64 in einem gesonderten Band angenommen. Das Buch richtet sich gleichermaßen an den Anfänger wie auch an den fortgeschrittenen Song- und Soundprogrammierer.

Nach Lektüre der ersten Seiten können dem Computer anhand von einfachen Routinen schon eine ganze Reihe von Tönen entlockt werden. Die Erstellung ein- oder mehrstimmiger Musikstücke wird, wie auch alle folgenden Beispiele, leicht verständlich beschrieben. Mehrere Tabellen über Tonfrequenzen, Filtereinstellungen, sämtliche Register des SID und ein kleines Lexikon zur Computermusik sind zum Nachschlagen geeignet. Leider sind die Tabellen nicht zusammengefaßt im Anhang, sondern nur in den jeweiligen Kapiteln zu finden. Wer sich nicht mehr mit Basic begnügt, kann die im zweiten Teil des Buches abgedruckten und kommentierten Assemblerlistings, oder auch die zugehörigen Basic-Lader ausprobieren.

Dieser Teil ist weiterhin mit wichtigen Erklärungen für die fortgeschrittene Programmierung versehen. Unter anderem werden die Vorteile behandelt, die Assembler, aber auch Basic bieten. Im Anhang erscheint ein Ausblick auf weitere Anwendungen, zum Beispiel der Anschluß einer externen Tastatur oder die Kombination des Computers mit einer Stereoanlage.

Kritisch ist anzumerken, daß der Autor Thomas Dachsel hier kräftig Werbung macht für sein Musikprogramm »Synthimat«.

Insgesamt gesehen ist das Buch sicherlich eine Bereicherung für alle, die die erstaunlichen Möglichkeiten des Musikus C 64 umfassend nutzen wollen. (Thomas Kraul/ev)

Info: Dachsel, Das Musikbuch zum Commodore 64, Data Becker, 200 Seiten, ISBN 3-89011-012-6, 34 Mark

Wordstar für die Sekretärin

Kein Irrtum: Seit es den C 128 gibt, gehören Buchbesprechungen über Wordstar auch ins 64'er-Magazin, denn jetzt ist das weitverbreitete Textverarbeitungssystem auch Commodore-Besitzern zugänglich. Wordstar zu loben ist sicher überflüssig, daß aber ein gutes Anleitungsbuch gerade für Wordstar mit seinen vielen Möglichkeiten wichtig ist, wird jeder zugeben, der in das Handbuch geblickt hat. Nicht nur der Sekretärin, sondern jedem Wordstar-Anfänger kann man das vorliegende Trainingsbuch mit gutem Gewissen empfehlen. Es bleiben kaum Fragen offen.

In Umkehrung eines bekannten Sprichwortes beginnt das Buch mit der Behauptung: »Aller Anfang ist nicht schwer«. Das Buch bringt den Beweis. Unabhängig von der vorhandenen Hardware lernt der Leser schrittweise mit der geplagten Sekretärin. Ihrem Chef kommen die guten Einfälle immer etwas verspätet. Mit Wordstar wird der weihnachtliche Brief an einen Geschäftspartner eingegeben, von Tippfehlern befreit und die erste Fassung auf Diskette gespeichert. Natürlich kommt es auch bei den nächsten Versionen des Briefes noch zu vielen Änderungen, die dank Wordstar und der guten Erklärung im Buch keine Schwierigkeiten machen.

Die Vorstellung weiterer Befehle aus dem Eröffnungsmenü, die Erläuterung der wichtigsten Punktbefehle und viele Tips runden den Inhalt ab. Selbstverständlich wurde vom Autor keine Vollständigkeit angestrebt, wer aber Wordstar mit diesem Buch als Einführung kennengelernt hat, dürfte keine Schwierigkeiten haben, später seine spezielle Anwendungsprobleme mit Hilfe des Handbuches zu lösen. (D. Hein/ev)

Info: Klaus Diehl, WordStar für die Sekretärin, Markt & Technik, 138 Seiten, ISBN 3-89090-128-X, Preis 29,80 Mark

Floppy VC 1541

Schon mancher Besitzer der Commodore-Floppy hat mit der 1541 Ärger gehabt. Zu anfällig ist die Mechanik. Reparaturen in der Werkstatt dauern lange und kosten viel Geld. So ist es kein Wunder, daß viele 1541-Besitzer versuchen, notwendige Pflege- und Reparaturarbeiten selbst durchzuführen. Daß Eingriffe in die Floppy nur nach Ablauf der Garantiezeit und von Leuten, die Erfahrung im Umgang mit hochempfindlichen Geräten haben, gemacht werden dürfen, kann nicht oft genug betont werden. Die vermeintliche Einsparung könnte sich leicht ins Gegenteil verkehren.

Die in den Kapiteln 1 und 2 beschriebenen Pflege- und Überwachungsarbeiten als Wartungsarbeiten können mit Hilfe der genauen Anleitungen und der vielen Zeichnungen im Buch von jedem, der nicht zwei linke Hände hat, ausgeführt werden. Das dritte Kapitel ist dem Justieren des Schreib-/Lesekopfes gewidmet. Sehr genau wird dabei geschildert, wie mit Hilfe des auf der Demo-Diskette befindlichen »Display T&S« geprüft werden kann, ob der Stoppanschlag für die Spur 1 richtig eingestellt ist und ob sich der Schreib-/Lesekopf über alle Spuren korrekt bewegt. Mit Hilfe des Buches muß man den Stoppanschlag gegebenenfalls um 0,1 bis 0,35 mm verstellen, noch schwieriger ist die Justage des Schreib-/Lesekopfes, obwohl das Buch genau beschreibt, wie man dabei mit oder ohne Oszilloskop vorgehen muß. Vier Programmlistings für Programme zum Erstellen einer Spureinstellungsdiskette, zur Spureinstellung, für einen Schreib-/Lesetest und zur Geschwindigkeitseinstellung können abgetippt werden und helfen dann bei den angeführten Arbeiten.

Für den Fachmann sehr nützlich ist die ausführliche Darstellung der Elektronik und Digitaltechnik. Dieses Kapitel zeugt allerdings von einem Mißverständnis: Wenn der Autor meint, der Leser müsse in diesem Buch zunächst über so primitive Voraussetzungen wie die Erklärung eines einfachen Stromkreises oder die Kennzeichnung von Widerständen informiert werden, versteht man nicht, welchen Sinn die vielen detaillierten Schaltpläne und Übersichten haben sollen. Dem Elektronik-Neuling nützen die letzten Kapitel nichts, dem Kenner müßte der Autor die Einführung ersparen.

Sollten etwa nur Seiten gefüllt werden? In der vorliegenden Form wird der unbestreitbare Wert der letzten Abschnitte ebenso wie durch die mangelhafte Rechtschreibung unnötig herabgesetzt. (D. Hein/ev)

Info: Reinhold Herrmann, Floppy VC 1541, Data Becker 1985, 220 Seiten, ISBN 3-89011-079-7, Preis 49 Mark

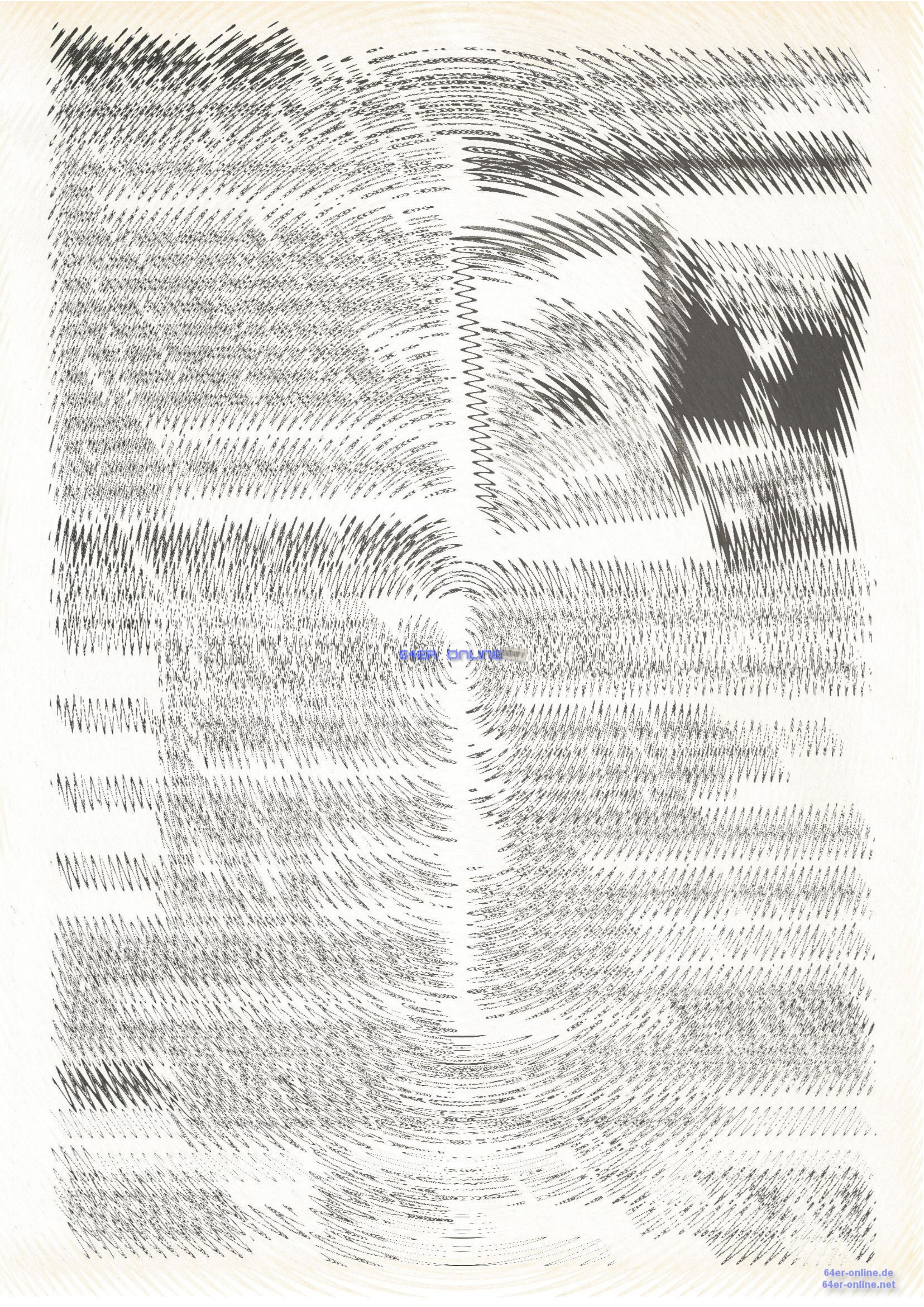
Das große Spielebuch

Die erste Hälfte des 141 Seiten umfassenden Buches bringt nach einer sehr knappen Erklärung der Spielidee 19 Programme zum Abtippen. Obwohl das Abtippen der Listings mit den vielen Steuerzeichen eigentlich eine Zumutung ist, erhält man als Lohn für diese Mühe dann 19 recht schöne Spiele. Es seien hier Spielautomat, Roulette, Irrgarten, magisches Quadrat und Konzentrationstest beispielhaft erwähnt. Leider hat es der Autor versäumt, den Programmen Anregungen für eigene Änderungen der Spiele anzufügen. Es bleibt damit für die erste Hälfte des Buches nur die Freude am richtig abgetippten Programm und dem Spiel.

Etwas völlig Unerwartetes bietet dann der zweite Teil des Buches: Wer würde schon in diesem Buch nach einer ausführlichen Anleitung für das Arbeiten mit hochauflösender Grafik und dem Tonteil des C 64 suchen? Obwohl der Inhalt dieser Kapitel dem Kenner nichts Neues bietet, kann ein Neuling sich hier doch gut über das Erstellen und Bewegen von Sprites sowie das Arbeiten mit dem Tonerzeuger unter Basic informieren.

(D. Hein/ev)

Info: Franz Ende, Das große Spielebuch — Commodore 64 —, Markt & Technik, 140 Seiten, ISBN 3-922120-63-6, Preis 29,80 Mark



Programmieren Sie

Strukturiert programmieren in Basic? vielen Beispielen, wie. Alle Beispiele damit Sie sehen, wie es in einer

Kann man auch in Basic strukturiert programmieren?

Natürlich kann man. Die oft gehörte Behauptung, man könne strukturiert nur in einer »strukturierten« Computersprache programmieren, ist Unsinn. Sie beruht auf einem schlichten Denkfehler, nämlich dem, daß Programmieren identisch sei mit dem Schreiben von Programmbefehlen.

Tatsächlich läuft (vernünftiges) Programmieren in zwei voneinander unabhängigen Schritten ab:
Schritt 1: Entwerfen eines Programms
Schritt 2: Codieren des Programmentwurfs

Ein Programm entwerfen heißt einen Plan für das Programm machen. Dieser Plan ist an dem Problem orientiert, das man mit dem Computer lösen will, nicht an der Programmiersprache, mit Hilfe derer man dem Computer seine Absichten mitzuteilen gedenkt. Das ist wie wenn Sie ein Bild aufhängen. Das planen Sie auch nicht, indem Sie über den Hammer nachdenken. Vielmehr geht es um das Bild und die Wand und andere Bilder und Möbelstücke, die an dieser Wand stehen. Zu diesem Zeitpunkt, da Sie sich überlegen, an welcher Stelle genau das Bild einmal hängen soll, ist der Hammer noch völlig ohne Belang. Der Hammer wird erst dann akut, wenn Sie daran gehen, Ihren Plan zu realisieren, das Bild tatsächlich aufhängen.

Der Hammer ist das Werkzeug, das Sie benutzen, um Ihren Plan in die Tat umsetzen, mehr nicht. Eine Computersprache ist das Werkzeug, das Sie brauchen, um einen Programmentwurf in ein Programm umzusetzen, mehr Beachtung verdient sie nicht. Das heißt sie kommt erst dann ins Spiel, wenn der Programmentwurf fertig ist und codiert werden soll. Und an dieser Stelle erst kann die Frage gestellt werden, was der Unterschied zwischen einer unstrukturierten Sprache wie Basic und einer strukturierten Sprache wie Pascal oder Elan oder Comal ist.

Der Unterschied ist der: Eine »strukturierte« Sprache stellt Befehle zur Verfügung, die es leichter machen, einen gut strukturierten Programmentwurf in ein Computerprogramm umzusetzen. Eine »unstrukturierte« Sprache wie Basic hat keine oder nur ein paar einzelne vorgefertigte Befehlsstrukturen dieser Art. Das heißt aber nicht, daß deshalb die Umsetzung eines strukturierten Programmentwurfs nicht möglich wäre; das bedeutet nur, daß wir uns in Basic entsprechende Befehlsstrukturen selber machen müssen. Mit einem Wort: Basic ist unbequemer, aber das ist auch schon alles.

Ein Beispiel, damit Sie sehen, was ich meine. Eine Schleife soll so lange durchlaufen werden, bis die Taste »E« gedrückt wird; dann soll die Schleife verlassen werden.

In einer strukturierten Sprache wie Comal läßt sich dies so programmieren (Listing 1).

```
10   REPEAT
20   ...
30   ...
30   ...
40   taste$=KEY$
50   UNTIL taste$ = "e"
```

Listing 1. Eine Schleife in Comal

In Basic haben wir keine REPEAT-UNTIL Struktur, müssen also selber eine erfinden (Listing 2).

```
10   rem repeat
20   ...
30   ...
40   ...
50   get taste$
60   if not (taste$="e") then 20
70   rem until
```

Listing 2. REPEAT...UNTIL in Basic

Wenn wir eine solche Befehlsstruktur einmal erfunden haben, dann steht sie dem Basic-Codierer genau so zur Verfügung wie eine vorfabrizierte dem, der in Comal oder Pascal codiert. Und das heißt, das Codieren eines strukturierten Programmentwurfs wird in Basic (fast) genau so bequem wie in Comal oder Pascal. In dieser Aufsatzserie wollen wir solche Befehlsstrukturen für Basic entwickeln.

Denken Sie, wenn auch vielleicht ungern, kurz einmal an den Deutschunterricht in Ihrer Schulzeit zurück. Da mußten Sie Aufsätze schreiben. Sie lernten, daß ein Aufsatz aus Einleitung, Hauptteil und Schluß besteht; daß jeder Hauptgedanke seinen eigenen Absatz wert ist; daß Gedanken in einer logischen Abfolge dargestellt werden sollten und außerdem in einer einfachen und klaren Sprache.

Was Sie da lernten, waren nichts als ein paar technische Tricks, die, wenn Sie sie benutzten, das Aufsatzschreiben leichter machten. Mit Hilfe dieser Tricks kamen Sie schneller, effektiver ans Ziel. Mit weniger Frust und mehr Spaß. Und hatten als Ergebnis einen Aufsatz, den der Leser verstand. Der Zweck des Aufsatzes, sich mitzuteilen, war erreicht.

Was heißt »strukturiert«, und warum überhaupt?

Programmieren ist dem Aufsatzschreiben recht ähnlich. Auch Programmieren soll schnell, mühelos, effektiv vonstatten gehen, und sein Ergebnis, das Programm, soll verständlich sein.

Wobei Sie sich zwei Adressaten gleichermaßen mitteilen wollen: Einmal natürlich dem Computer, der aufgrund des Programms haargenau das tun soll, was Sie sich ausgedacht haben; zum andern wollen Sie sich dem menschlichen Leser mitteilen. Sie meinen, wer könnte schon Ihr Programm lesen wollen? Vor allem Sie selbst!

Nichts ist so gut, als daß es nicht verbessert werden könnte, um mit dem Dichter zu sprechen. Wenn dieser Spruch überhaupt auf etwas zutrifft, dann auf Computerprogramme. Wer programmiert, weiß es. Wenn Sie aber ein Computerprogramm verbessern wollen, dann müssen Sie es so geschrieben haben, daß Sie's auch in einem halben Jahr noch verstehen.

Schreiben Sie also Ihre Programme mit Einleitung, Hauptteil und Schluß; machen Sie Absätze und ordnen Sie diese logisch an; benutzen Sie eine einfache, klare Ausdrucksweise. Mit andern Worten, programmieren Sie strukturiert.

Welche technischen Tricks Ihnen dabei helfen können, darum soll es in dieser Serie gehen.

Ein Computerprogramm besteht aus Bausteinen.

Es gibt Bausteine verschiedenster Komplexität. Einfache Bausteine sind zum Beispiel Befehlswörter wie »Print« oder »Input«, oder Daten wie »3.14« oder »pi«. Schon komplexer sind Befehlssätze wie »Print pi« oder »Input "Bitte geben Sie Ihren Namen ein"; name$«. Noch komplexer sind Befehlsstrukturen, wie sie oben illustriert wurden (Listings 1 und 2); sie sind vergleichbar den Absätzen in einem menschensprachlichen Text.

In dieser Serie geht es um solche komplexen Bausteine. Und zwar werden wir uns mit drei Bausteintypen beschäftigen:

1. den Steuerbausteinen — mit ihrer Hilfe wird der Programmfluß gelenkt
2. den Unterprogrammbausteinen — sie dienen dazu, zusammengehörige Befehle zusammenzufassen
3. den Modulbausteinen — aus diesen besteht schließlich das Programm

In diesem ersten Teil werden die Steuerbausteine behandelt; im nächsten geht es um Unterprogramme; zum Schluß wollen wir ein Programm bauen und dabei die Modulbausteine in den Mittelpunkt stellen.

Zunächst aber lassen Sie uns ein paar Grundregeln für die Codierung von Programm-Bausteinen zusammenstellen.

Grundregeln für Programmbausteine

Die Bausteine, als denen Programmentwürfe zusammengesetzt sind, sind Gefäße für gedanklich zusammenhängende Einheiten. Im Programm werden sie deshalb als zusammenhängende Zeilenblocks realisiert (Bild 1).

strukturiert! (Teil 1)

Kein Problem. Wir zeigen Ihnen mit werden auch in Comal geschrieben, »strukturierten« Sprache aussieht.

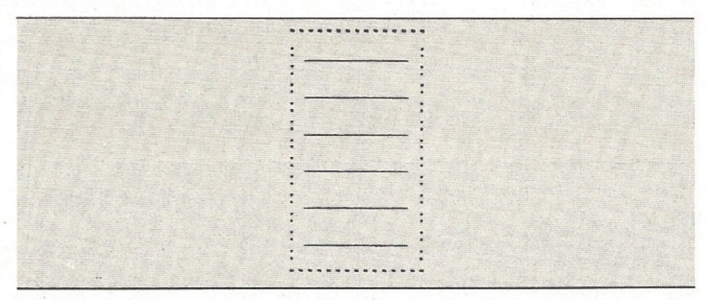


Bild 1. Ein Zeilenblock besteht aus einer oder mehreren Programmzeilen

Diese Zeilenblocks sind nach folgenden Regeln gebaut:
1. Ein Block besteht aus einer oder mehreren Programmzeilen. Dabei enthält jede Programmzeile nur einen Befehl.
2. Jeder Block hat nur einen Eingang und nur einen Ausgang. Der Eingang befindet sich in der ersten Zeile des Blocks, der Ausgang in der letzten Zeile.
3. Der GOTO-Befehl, soweit er überhaupt verwendet wird (in Schleifen- und in Verzweigungsblocks), darf nur zu Sprüngen innerhalb eines Blocks benutzt werden, und auch dabei gelten bestimmte Regeln.
4. Es kann manchmal nützlich und sinnvoll sein, einen Zeilenblock in eine Einzelzeile umzuwandeln (Bild 2).

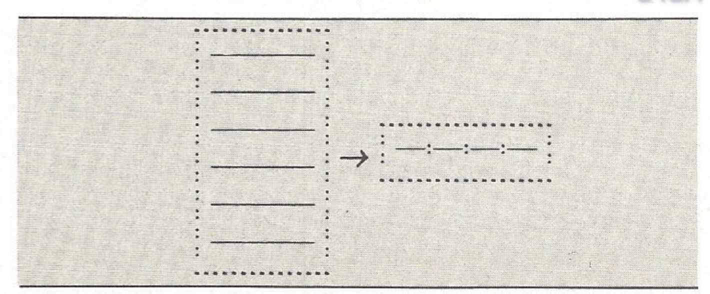


Bild 2. Manchmal ist es vorteilhaft, aus mehreren Zeilen eine zu machen.

Einzeilenblocks sind in vielen Fällen mit weniger Aufwand codierbar, übersichtlicher und leichter verständlich und meist schneller in der Programmausführung.

Unabdingbar ist bei einer derartigen Umwandlung allerdings, daß der gesamte Block, der ganze Baustein, in einer Zeile Platz findet, so daß die gedankliche Einheit des Bausteins nicht zerstört wird.

Wenn diese Regeln beachtet werden, dann werden dadurch automatisch Spaghetti-Programme vermieden, die, wie jeder weiß, der solche gemacht hat, das Leben des Programmierers sehr erschweren.

Die Steuerbausteine

Wir werden die folgenden drei Typen von Steuerbausteinen besprechen:
1. die Abfolge oder Sequenz
2. die Schleife
3. die Verzweigung

Mehr als diese drei Typen braucht man nicht, mit diesen drei Typen läßt sich jedes Steuerungsproblem lösen.

Die Steuerbausteine unterscheiden sich in der Art und Weise, wie sie den Programmfluß lenken: In einer Sequenz arbeitet das Programm linear einen Befehl nach dem andern ab; die Schleife bewirkt, daß eine Befehlsreihe immer wieder von neuem durchlaufen wird; Verzweigungen machen es möglich, das Programm auf alternative Wege zu schicken, wobei die Wahl des Wegs von bestimmten Bedingungen abhängig sein kann.

Wir untersuchen diese drei Bausteintypen der Reihe nach. Dabei werden wir feststellen, daß es bei Schleifen und Verzweigungen Untertypen gibt, die jeweils für verschiedene Einsatzbedingungen geeignet sind. Diese sollen beschrieben und anhand kleiner Programmierprobleme illustriert werden.

Zuerst machen wir jeweils einen Programmentwurf, der die gedankliche Logik des Bausteins zeigt. Dann kommt der zweite Schritt, die Codierung. Wo Basic keine passende Befehlsstruktur zur Verfügung stellt, werden wir selber eine entwickeln.

Dabei lassen wir uns anregen. Man soll ja, heißt es, das Rad nicht immer wieder neu erfinden. Die Anregung holen wir vor allem aus Comal. Diese Programmiersprache enthält eine Reihe sehr gut durchdachter Befehlsstrukturen, die in der Regel so aufgebaut sind, daß die Bausteine direkt, ohne geistige Umwege, das heißt der menschlichen Logik folgend, umgesetzt werden können. Diese Befehlsstrukturen können als Vorbild für unsere eigenen Basic-Befehlsstrukturen dienen.

Wir wollen deshalb unsere Programmentwürfe jeweils zuerst in Comal codieren und dann versuchen, ähnliches in Basic zu realisieren.

In einigen Fällen werden zusätzlich noch alternative Codierungsmöglichkeiten aufgezeigt, weil sie vielleicht schneller sind, oder übersichtlicher, oder bequemer zu codieren.

Was Comal angeht, so codieren wir meist in der (fast) kostenlosen Diskettenversion 0.14. Wo einmal die Steckmodulversion 2.01 benutzt wird, wird ausdrücklich darauf hingewiesen. (Bezugsinformation für Comal am Schluß dieser Folge.)

Die Abfolge oder Sequenz

Der einfachste Steuerbaustein ist die Abfolge oder Sequenz von Befehlen, die so, wie sie dastehen, hintereinander abgearbeitet werden. Eine spezielle Befehlsstruktur gibt es dafür weder in Basic noch in Comal. (Bild 3: Dollarrechner).

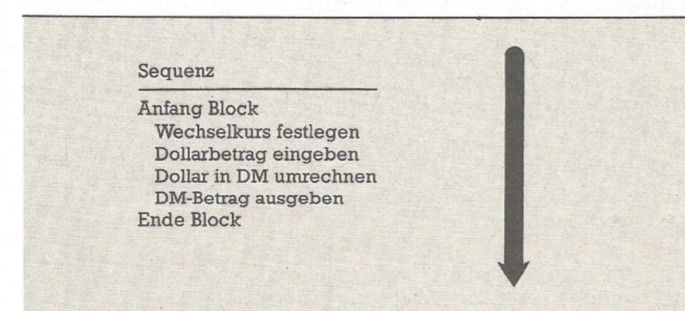


Bild 3. Die Sequenz

Codiert in Comal als Sequenzblock sieht dieser Baustein so aus (Listing 3).

```
0010    sequenz
0020    kurs:=3.21
0030    Input "Bitte Dollarbetrag eingeben:":dollarbetrag
0040    mark:=dollarbetrag*kurs
0050    Print mark
0060    ende sequenz
```

Listing 3. Die Sequenz in Comal

Die Basic-Version dieses Bausteins ist fast identisch; wobei zu beachten ist, daß Basic bei Variablennamen nur die beiden ersten Zeichen berücksichtigt, es würde also genügen, KU, DO und MA zu schreiben. Aber das macht die Basic-Codierung natürlich schwerer verständlich (Listing 4).

```
10   rem sequenz
20   :kurs=3.21
30   :input "Bitte Dollarbetrag eingeben"; dollarbetrag
40   :mark=dollarbetrag*kurs
50   :print mark
60   rem ende sequenz
```

Listing 4. Die Sequenz in Basic

Hinweis: Das Einrücken von Zeilen kann in Commodore-Basic auf zweierlei Weise erreicht werden. Einmal durch Verwendung von Doppelpunkten wie in Listing 4. Zum andern so: Nach der Zeilennummer geben Sie irgendein Grafikzeichen ein (zum Beispiel SHIFT N), dann so viele Leerzeichen, wie Sie brauchen, dann den Befehl, der in der Zeile stehen soll. Das Grafikzeichen verschwindet, wenn Sie die Zeile listen. Aber Vorsicht: Wenn Sie später in der Zeile eine Änderung vornehmen, geht das (unsichtbare) Grafikzeichen verloren, und der Zeilentext rückt wieder nach links. Sie müssen dann die Zeile neu »formatieren«.

Die Schleifen

Der Bausteintyp Schleife wird dann verwendet, wenn eine Reihe von Befehlen mehrmals hintereinander wiederholt werden soll. Dabei sind mehrere Arten von Schleifen zu unterscheiden:

1. die Zählschleife
2. die WHILE-Schleife
3. die UNTIL-Schleife
4. die LOOP-Schleife
5. die Endlosschleife

Die Zählschleife benutzen wir immer dann, wenn wir von vornherein wissen oder ausrechnen können, wie oft die Schleife durchlaufen werden soll (siehe Bild 4).

Dieser Baustein soll die einzelnen Buchstaben eines Texts senkrecht untereinander schreiben. Die Anzahl der Schleifendurchgänge hängt von der Länge des Texts ab und läßt sich also ausrechnen. Der Text wird dem Baustein mitgegeben.

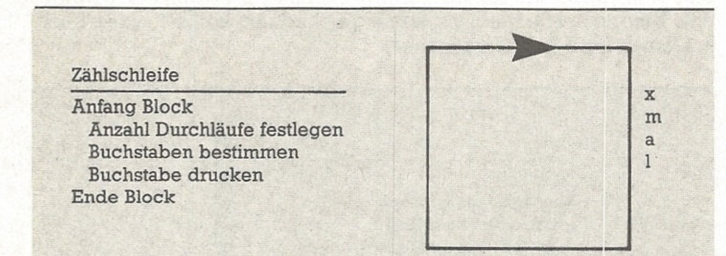


Bild 4. Die Zählschleife

Für Zählschleifen stellen sowohl Comal wie Basic eine (fast identische) Befehlsstruktur zur Verfügung (Listing 5).

```
0010   FOR position:=1 TO LEN(text$) DO
0020     buchstabe$:=text$(position)
0030     PRINT buchstabe$
0040   ENDFOR position

10     for ps=1 to len(text$,ps,1)
20       buchstabe$=mid$(text$,ps,1)
30       print buchstabe$
40     next ps
```

Listing 5. Die Zählschleife in Comal (oben) und Basic (unten)

Die Befehlsfolge innerhalb des Schleifenrahmens können wir zusammenfassen zu einem einzigen Befehl (siehe Listing 6), was uns die Möglichkeit gibt, den gesamten Block in einer einzigen Zeile unterzubringen (wodurch die Schleife schneller durchlaufen wird).

Oft weiß man nicht von vornherein und kann es auch nicht ausrechnen, wie oft eine Befehlsfolge durchlaufen werden muß, sondern die Anzahl der Durchläufe hängt von unvorhersehbaren Situationsbedingungen ab, zum Beispiel davon, ob eine Variable schon einen bestimmten Wert überschritten hat. Diese Situationsbedingung muß regelmäßig überprüft werden. Je nachdem, wann die Überprüfung

```
Der Einzeilenblock in Comal
  0010 FOR position:=1 TO LEN(text$) DO PRINT text$(position)
Der Einzeilenblock in Basic
  10 for ps=1 to len(text$) :buchst$=mid$(text$,ps,1) :print buchst$
:next
```

Listing 6. Der Einzeilenblock für Listing 5

stattfindet, ob vor dem Schleifendurchlauf oder danach oder mittendrin oder überhaupt nicht, ergeben sich verschiedene weitere Schleifentypen.

Bei der WHILE-Schleife wird zu Beginn überprüft, ob sie durchlaufen werden soll.

Ein Text wird bei jedem Schleifendurchgang um einen Buchstaben kürzer. Die Schleife wird nur dann durchlaufen, wenn überhaupt ein Text vorhanden ist (siehe Bild 5 und Listing 7).

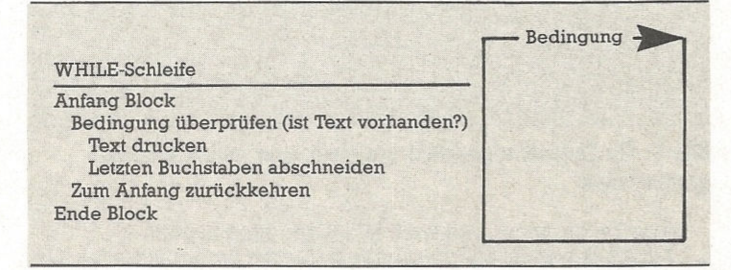


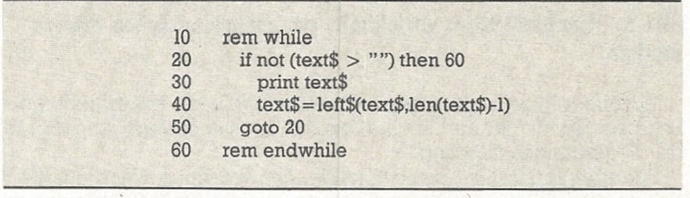
Bild 5. Die WHILE-Schleife

```
0010   WHILE text$ > "" DO
0020   PRINT text$
0030   text$:=text$(1:LEN(text$)—1)
0040   ENDWHILE
```

Listing 7. Die WHILE-Schleife in Comal

Wenn dem Schleifenblock ein Text mitgegeben wurde, dann wird er so lange wiederholt, bis die Textlänge 0 geworden ist; wenn von vornherein kein Text vorhanden ist, findet überhaupt kein Schleifendurchlauf statt.

In Basic gibt es für die WHILE-Schleife (und dies gilt auch für die folgenden Schleifentypen) keine spezielle Befehlsstruktur, so daß wir eine Blockstruktur dafür erfinden müssen. Diese soll so beschaffen sein, daß sie die Logik des Bausteins möglichst genau abbildet; die Comal-Struktur wird als Anregung mitbenutzt (Listing 8).



```
10   rem while
20     if not (text$ > "") then 60
30       print text$
40       text$=left$(text$,len(text$)-1)
50     goto 20
60   rem endwhile
```

Listing 8. Die WHILE-Schleife in Basic

Das Programm bleibt so lange in diesem Block, wie ein Text vorhanden ist. Wenn die Bedingung, daß ein Text vorhanden ist, nicht mehr gilt, wird der Block verlassen, und zwar durch den einzigen Ausgang, der vorgesehen ist, also durch die Zeile, die die Ende-Markierung enthält. Dies ist eine REM-Zeile. REM-Zeilen sind bei dieser Art von Codierung oft integraler Bestandteil des Programms und dürfen dann auf keinen Fall gelöscht werden!

Bitte beachten Sie besonders, daß bei der Basic-Version der WHILE-Schleife die Bedingung für den Schleifendurchlauf mit NOT negiert werden muß (dies werden wir noch öfter finden). Das kann

```
20   if text$ "" then print text$ :
       text$=left$(text$,len(text$)-1) : goto 20
```

Listing 9. Die WHILE-Schleife in Basic als Einzeiler

zu Denkschwierigkeiten führen und ist vielleicht dadurch fehleranfällig. Man kann diesem Problem jedoch entgehen, wenn man versucht, WHILE-Schleifen in einer einzigen Zeile zu codieren (Listing 9).

Diese Zeile wird nur ausgeführt, wenn die Bedingung zu Beginn zutrifft. Wenn zu viele Befehle auszuführen sind, so daß die Zeile nicht ausreicht, kann man die ganze Befehlsreihe als Unterprogramm auslagern. Die Codierung könnte dann so aussehen (Listing 9a):

```
20    if text$ > ""then gosub 1000 : goto 20
90    end
1000  rem proc:text kürzen
1010       print text$
1020       text$=left$(text$,len(text$)-1
1030  return
```

Listing 9a. Die WHILE-Schleife mit Unterprogrammaufruf

Übrigens kennt auch Comal einzeilige WHILE-Schleifen, sofern nur ein einziger Befehl auszuführen ist. Listing 9b entfernt überzählige Leerzeichen zu Beginn einer Zeichenkette.

```
0010    WHILE text$(1)=" " DO text$:=text$(2:LEN(text$))
```

Listing 9b. Die WHILE-Schleife in Comal als Einzeiler

Können Sie dasselbe in einer Basic-Zeile ausdrücken? Denken Sie daran, daß einzeilige WHILE-Schleifen in Basic mit IF beginnen und mit einem GOTO-Befehl auf dieselbe Zeile enden.

WHILE-Schleifen sind immer dann angebracht, wenn man sicherstellen will, daß die Schleife eventuell ganz übersprungen wird. Alle anderen Schleifentypen (mit Ausnahme der LOOP-Schleife, die einen speziellen Fall darstellt) werden mindestens einmal durchlaufen. Das gilt für die Zählschleife ebenso wie für die Endlosschleife und die UNTIL-Schleife, der wir uns nun als nächstes zuwenden.

Die UNTIL-Schleife überprüft die Situationsbedingungen erst am Ende eines Durchlaufs und entscheidet dann, ob ein weiterer notwendig ist (siehe Bild 6 und Listing 10).

DATA-Zeilen werden so lange gelesen, bis ein bestimmter Suchbegriff gefunden ist. Dann wird die Schleife verlassen.

```
UNTIL-Schleife
Anfang Block
  Datum lesen
  Bedingung überprüfen
  (Suchbegriff gefunden?)
  und ev. zurück an Anfang
Ende Block
                                  Bedingung
```

Bild 6. Die UNTIL-Schleife

```
Comal:  0010  REPEAT
        0020    READ text$
        0030  UNTIL text$=gesucht$

Basic:  10   rem repeat
        20     read text$
        30     if not (text$=gesucht$) then 20
        40   rem until
```

Listing 10. Die UNTIL-Schleife in Comal (oben) und Basic (unten)

Die Bedingung wird auch hier wieder mit NOT negiert. Leichter zu codieren ist, ähnlich wie bei der WHILE-Schleife, ein Einzeilenblock (Listing 11).

```
20    read text$ : if not (text$=gesucht$) then 20
```

Listing 11. Basic-Einzeiler für die UNTIL-Schleife

Kennzeichnend für UNTIL-Schleifen dieser Art ist IF-NOT-THEN als letzte Befehlsgruppe und ein Sprung zum Beginn der Zeile.

Es gibt noch eine weitere Möglichkeit, UNTIL-Schleifen in Basic zu codieren, allerdings in etwas unüblicher Weise (dafür ist die Schleife jedoch schneller). Wir benutzen dafür den Zählschleifenbefehl FOR-NEXT (Listing 12).

```
10   rem repeat
20     for i=0 to 1
30       read text$
40       i=abs(text$=gesucht$)
50     next i
60   rem until
```

Listing 12. Die UNTIL-Schleife in Basic mit FOR...NEXT

Das funktioniert so: Die Schleife wird auf jeden Fall einmal durchlaufen. Dann wird in Zeile 40 überprüft: (TEXT$=GESUCHT$). Dieser Boolesche Ausdruck ergibt den Wert 0, wenn er falsch ist, das heißt wenn der gelesene Text noch nicht mit dem gesuchten Text übereinstimmt. In diesem Fall erhält I in Zeile 40 den Wert 0, und wenn das Program zu NEXT kommt, erfährt er es, daß I den Endwert nocht nicht erreicht hat und deshalb ein weiterer Durchlauf erforderlich ist.

Dies geht so lange, bis der richtige String gefunden ist. Dann hat (TEXT$=GESUCHT$) den Wert —1; durch die Funktion ABS wird das Minuszeichen entfernt, und I wird zu 1. Wenn aber I=1 ist, dann ist die Aufgabe der Schleife erfüllt und der ganze Block wird verlassen.

Übrigens entsteht ein »OUT OF DATA«-Fehler, wenn die gesuchte Zeichenkette gar nicht unter den Daten vorhanden ist. Deshalb muß normalerweise noch eine andere Bedingung geprüft werden, zum Beispiel, ob (TEXT$="ENDE") ist. Dies ist ohne Schwierigkeiten codierbar, es muß nur die Zeile 40 in folgender Weise geändert werden (Listing 12a):

```
40    i=abs(text$=gesucht$ or text$="ende"
```

Listing 12a. Sicherheitsmaßnahme gegen OUT OF DATA

In derselben Weise sind die Bedingungsausdrücke in den anderen Beispielen zu ändern.

Auch diese Codierungsart kann im übrigen in einer Zeile geschehen (Listing 13).

```
20   for i=0 to 1 : read text$ : i=abs(text$=gesucht$ or text$="ende":
     next
```

Listing 13. Listing 12 und 12a als Einzeiler

UNTIL-Schleifen diesen Typs erkennt man daran, daß sie von 0 bis 1 zählen und im vorletzten Befehl den Wert der Zählvariablen (hier I) berechnen. Da sie auf diese Weise genügend deutlich markiert sind, können wir uns die beiden REM-Zeilen in Listing 12 sparen.

Die am häufigsten benötigte Schleifenart ist die UNTIL-Schleife, danach kommt die WHILE-Schleife, am seltensten besteht ein Bedürfnis für die LOOP-Schleife, der Schleife mit Abbruch. Bei der LOOP-Schleife wird die Situation während des Schleifendurchlaufs, also innerhalb der Schleife, überprüft; wenn die Prüfbedingung zutrifft, wird der gerade stattfindende Durchlauf abgebrochen, wie das nächste Beispiel illustriert (Bild 7).

Die Schleife wird so lange durchlaufen, bis eine befriedigende Antwort eingeht; wenn dies der Fall ist, wird abgebrochen.

```
LOOP-Schleife
Anfang Block
  Eingabe
  Bedingung überprüfen und
  eventuell zum Schleifen-Ende gehen       Bedingung
  Fehlerhinweis geben
  Zurück zum Anfang
Ende Block
```

Bild 7. Überprüfen einer Eingabe mit der LOOP-Schleife

```
0010    LOOP
0020      INPUT "Bitte geben Sie JA oder NEIN ein": antwort$
0030      EXIT WHEN antwort$="JA"OR antwort$="NEIN"
0040      PRINT "Bitte nur JA oder NEIN eingeben!"
0050    ENDLOOP
```

Listing 14. Bild 7, die LOOP-Schleife, mit Comal realisiert

Die LOOP-Schleife, die übrigens nur in der Comal-Version 2.01 auf diese Weise codierbar ist (Listing 14), überprüft, wie gesagt, die Situation inmitten des Schleifendurchlaufs; das heißt dieser Schleifentyp ist nur dann gerechtfertigt, wenn sowohl vor der Überprüfung als auch danach Anweisungen ausgeführt werden können.

```
10   rem loop
20     input "Bitte geben Sie JA oder NEIN ein'; antw$
30     if antw$="JA" or antw$="NEIN" then 60
40     print 'Bitte nur JA oder NEIN eingeben!"
50     goto 20
60   rem endloop
```

Listing 15. Die LOOP-Schleife in Basic

Das Beispiel illustriert übrigens ein typisches Einsatzgebiet für die LOOP-Schleife: Mit ihrer Hilfe kann man den Notstand proben, einfacher ausgedrückt: voraussehbare Fehler abfangen. Die LOOP-Schleife wird auf jeden Fall »halb«, das heißt bis zur Abbruchstelle, durchlaufen. Fortsetzung und Neuanfang der Schleife geschehen nur dann, wenn der vorausgeahnte Notstand eintrifft (Listing 15 = Basic-Version).

Gelegentlich benötigt man auch einen letzten Schleifentyp, die Endlosschleife, zum Beispiel bei einem Demonstrationsprogramm, das pausenlos immer wieder von neuem ablaufen soll (Bild 8 und Listing 16).

Dafür gibt es keine spezielle Befehlsstruktur, was aber auch nicht erforderlich ist, da sich Endlosschleifen ohne Mühe codieren lassen.

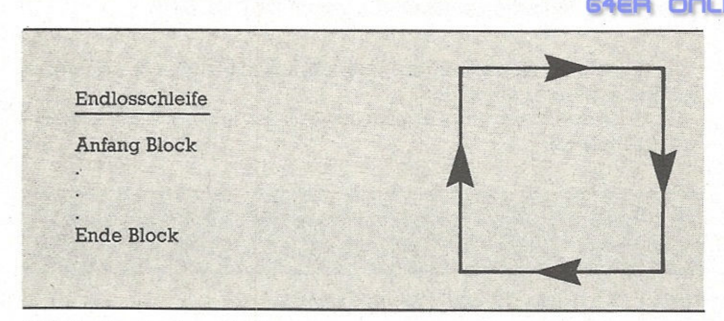


Bild 8. Die Endlosschleife

```
0010  REPEAT                    0010  LOOP
0020  ...                       0020  ...
0030  ...                       0030  ...
0040  ...                       0040  ...
0050  UNTIL TRUE=FALSE          0050  ENDLOOP

          10    rem endlos
          20      ...
          30      ...
          40      ...
          50      goto 20
          60    rem ende endlos
```

Listing 16. Die Endlosschleife in Comal 0.14 (links oben), Comal 2.01 (rechts oben) und in Basic (unten)

Die Verzweigungen

Befehlssequenzen weisen den Computer an, eine Reihe von Befehlen hintereinander abzuarbeiten; Schleifen veranlassen ihn, eine Reihe von Befehlen unter bestimmten Umständen mehrmals zu durchlaufen; der dritte Bausteintyp, die Verzweigung, ermöglicht es, den Computer mal zur einen, mal zu einer anderen Befehlsgruppe zu schicken. Wohin er geht, hängt von der jeweils herrschenden Situation ab.

Wir können drei Untertypen der Verzweigung unterscheiden:
1. den Abstecher
2. die Gabelung
3. die Mehrfachverzweigung (oder Kreuzung)

Die einzelnen Verzweigungstypen unterscheiden sich in der Art und/oder Anzahl der alternativen Wege, zwischen denen zu entscheiden ist.

Einen »Abstecher« machen, heißt im Leben, den eigentlichen Weg, auf dem man sich befindet, kurz zu verlassen, etwas zu erledigen, um dann wieder auf den Hauptweg zurückzukehren. Dazu muß natürlich ein spezieller Grund vorhanden sein. Beispielsweise, man ist auf dem Nachhauseweg, verspürt plötzlich Appetit auf Kaffee, tritt ins Kaffegeschäft, das am Weg liegt, trinkt die Tasse Kaffee und setzt anschließend den Heimweg fort. Es kann aber auch sein, daß man keinen Kaffeedurst hat, dann beachtet man das Kaffeegeschäft überhaupt nicht und bleibt unbeirrt auf dem Heimweg.

Dasselbe gilt für Computerprogramme. Unter einer bestimmten Bedingung wird das Hauptprogramm kurz unterbrochen, eine Befehlsreihe wird ausgeführt, dann läuft das Programm auf dem Hauptweg weiter.

Ein Name wird in Klein/Großschreibung eingegeben. Wenn der erste Buchstabe klein geschrieben wurde, wird er in einen Großbuchstaben umgewandelt; wenn er schon groß ist, findet kein Abstecher statt (Bild 9 und Listing 17).

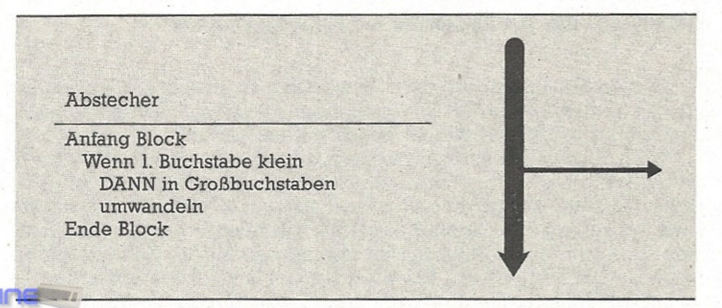


Bild 9. Großschreibung

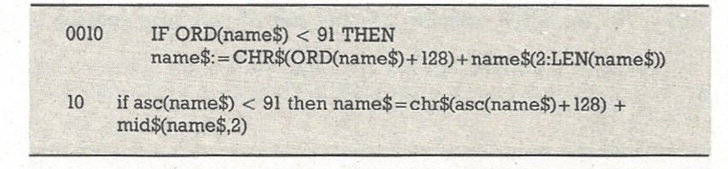
```
0010    IF ORD(name$) < 91 THEN
0020      ALT:=ord(name$)
0030      neu:=alt+128
0040      name$:=chr$(neu)+name$(2:LEN(name$))
0050    ENDIF

10      rem if
20        if not(asc(name$) < 91) then 60
30          alt=asc(name$)
40          neu=alt+128
50          name$=chr$(neu)+mid$(name$,2)
60      rem endif
```

Listing 17. Die Verzweigung als Abstecher in Comal und Basic

Wie früher schon, müssen auch hier die Bedingungen negieren, damit der natürliche Gedankenfluß erhalten bleibt.

Oft wird beim Abstecher allerdings nur ein einziger Befehl ausgeführt; dann können wir den ganzen Block in einer Zeile unterbringen, was die Verständlichkeit auch in Basic erhöht, schon deshalb, weil wie auch woanders, die Bedingung in ihrer »Grundform« verwendet werden kann. Dies sei am obigen Beispiel gezeigt, indem wir die Befehlsreihe zu einem einzigen Befehl zusammenfassen (Listing 17a).



```
0010    IF ORD(name$) < 91 THEN
        name$:=CHR$(ORD(name$)+128)+name$(2:LEN(name$))

10    if asc(name$) < 91 then name$=chr$(asc(name$)+128) +
      mid$(name$,2)
```

Listing 17a. Listing 17 als Einzeiler (oben: Comal, unten: Basic)

In Basic kann dem THEN übrigens nicht nur ein Befehl folgen, sondern mehrere — so weit halt in der Zeile Platz bleibt. Wie immer, kann man aber auch eine größere Anzahl von Befehlen unterbrin-

gen, indem man sie als Unterprogramm auslagert (siehe Listing 9).

Eine »Gabelung« in einem Programm ist wie eine Weggabelung: Es gibt zwei Möglichkeiten weiterzugehen, und es muß entschieden werden, welchen der beiden Wege man nimmt (Bild 10).

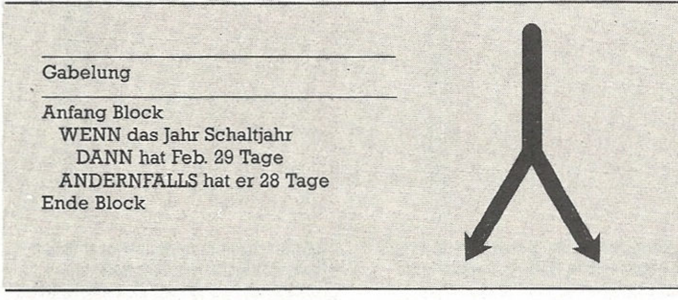


Bild 10. Die Verzweigung als Gabelung

```
0010  IF jahr MOD 4=0 THEN       10  rem if else
0020     tage:=29                20     if not(jahr/4=int(jahr/4)) then 50
0030  ELSE                       30       tage=29
0040     tage:=28                40     goto 60
0050  ENDIF                      50       tage=28
                                 60  rem endif
```

Listing 18. Die »Gabelung« in Comal und in Basic

(Der Operator MOD errechnet den Divisionsrest; Beispiel: 18 MOD 4 ergibt den Rest 2.)

Die von Comal zur Verfügung gestellte Befehlsgruppe IF=THEN-ELSE erlaubt die Codierung analog zur menschlichen Logik. In Basic müssen wir, um diese Logik zu erhalten, die Bedingung wiederum mit NOT negieren (Listing 18).

Und wieder gibt es die Möglichkeit, diese Negierung zu vermeiden, indem wir versuchen, den Gabelungsblock in einer einzigen Zeile unterzubringen (Listing 19).

```
20 tage=28 :   if jahr/4=int(jahr/4) then tage=29
```

Listing 19. Listing 18 als Einzeiler

Kommen wir zum letzten Verzweigungstyp, der Mehrfachverzweigung. Sie funktioniert wie eine Wegekreuzung, an der wir die Wahl zwischen drei oder mehr Alternativen haben (Bild 11 und Listing 20).

Für jeden Monat soll die entsprechende Anzahl von Tagen ermittelt werden können.

```
Mehrfachverzweigung
Anfang Block
  WENN der 1., 3., 5., 7., 8., 10., 12. Monat
    DANN 31 Tage
  WENN HINGEGEN der 4., 6., 9., 11. Monat
    DANN 30 Tage
  WENN HINGEGEN der 2. Mt.
    DANN 28 Tage
  ANDERNFALLS Fehlermeldung
  und 0 Tage
Ende Block
```

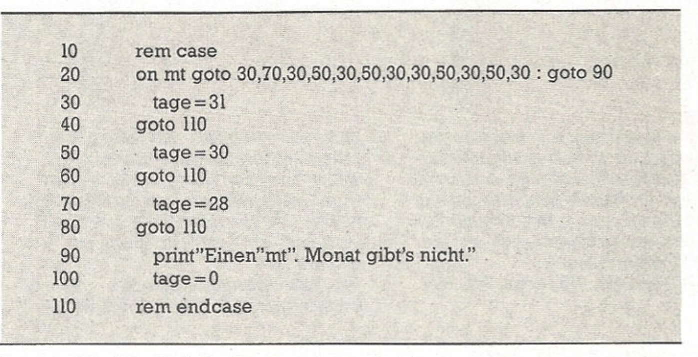
Bild 11. Die Mehrfachverzweigung

```
0010  CASE monat OF
0020  WHEN 1,3,5,7,8,10,12
0030    tage:=31
0040  WHEN 4,6,9,11
0050    tage:=30
0060  WHEN 2
0070    tage:=28
0080  OTHERWISE
0090    PRINT "Einen";monat;". Monat gibt's nicht."
0100    tage:=0
0110  ENDCASE
```

Listing 20. Die Mehrfachverzweigung in Comal

Die Case-Struktur funktioniert so: Es wird zuerst der Wert der Variablen MONAT überprüft. Angenommen, es handelt sich um den Juni, also den 6. Monat, dann geht das Programm in den Zweig, der in seiner Werteliste den Wert 6 enthält, also zur Zeile 40, und arbeitet die dazugehörigen Befehle ab; so wird TAGE zu 30. Danach wird der Block verlassen. Wenn der Wert von MONAT in keiner der Wertelisten enthalten ist, das heißt wenn MONAT kleiner als 1 oder größer als 12 ist, dann verzweigt das Programm auf OTHERWISE.

In Basic kann dieser (spezielle) Fall so codiert werden (Listing 21):



```
10    rem case
20    on mt goto 30,70,30,50,30,50,30,30,50,30,50,30 : goto 90
30      tage=31
40    goto 110
50      tage=30
60    goto 110
70      tage=28
80    goto 110
90      print"Einen"mt". Monat gibt's nicht."
100     tage=0
110   rem endcase
```

Listing 21. Die Mehrfachverzweigung in Basic

Die ON-Struktur in Basic (Zeile 20) überprüft, so wie dies auch bei CASE der Fall war, den Wert der Variablen MT. Wenn dieser Wert 6 ist, wählt das Programm die sechste Zeilennummer der Liste aus, also 50, und springt zu dieser Zeile. TAGE erhält dort den Wert 30, und das Programm verläßt den Block über die letzte Zeile. Wenn MT einen Wert hat, zu dem es keine Zeilennummer in der Liste gibt (0 oder größer als 12), springt das Programm zum nächsten Befehl, also zu GOTO 90. Eine negative Zahl führt zu einer Fehlermeldung.

Die Variable MT kann bei diesem Beispiel die Werte 1-12 annehmen, das heißt die Werte einer Zahlenfolge. Dies ist auch der einzige Fall, wo diese Struktur in Basic verwendet werden kann.

Die CASE-Struktur in Comal kennt diese Einschränkung nicht. Zum einen kann die Variable zu jedem Typ gehören, kann also auch eine Stringvariable sein. Zum anderen brauchen die Werte, welche die Variable annimmt, nicht Elemente einer Folge sein. Schließlich sind bei Zahlenvariablen auch negative Werte legal. Die Case-Struktur ist also in weit mehr Situationen einsetzbar als die mit Hilfe von ON imitierte Struktur in Basic.

Wenn die Bedingungsverhältnisse nicht einer Zahlenfolge entsprechen oder überhaupt komplexer sind, müssen wir eine andere Struktur benutzen, und zwar eine, die von der Comal-Struktur IF-ELIF-ELSE abgeleitet ist. (ELIF ist zusammengezogen aus ELSE IF).

Aufgabe: Ein Text soll verändert werden. Der Mehrfachverzweigungsblock bearbeitet jeweils ein Zeichen (Bild 12 und Listing 22/23).

```
Mehrfachverzweigung
Anfang Block
  WENN Vokal
    DANN ersetzen durch '*'
  WENN HINGEGEN Konsonant
    DANN in Großbuchstaben umwandeln
  WENN HINGEGEN Zahl
    DANN weglassen
  ANDERNFALLS ohne Veränderung übernehmen
Ende Block
```

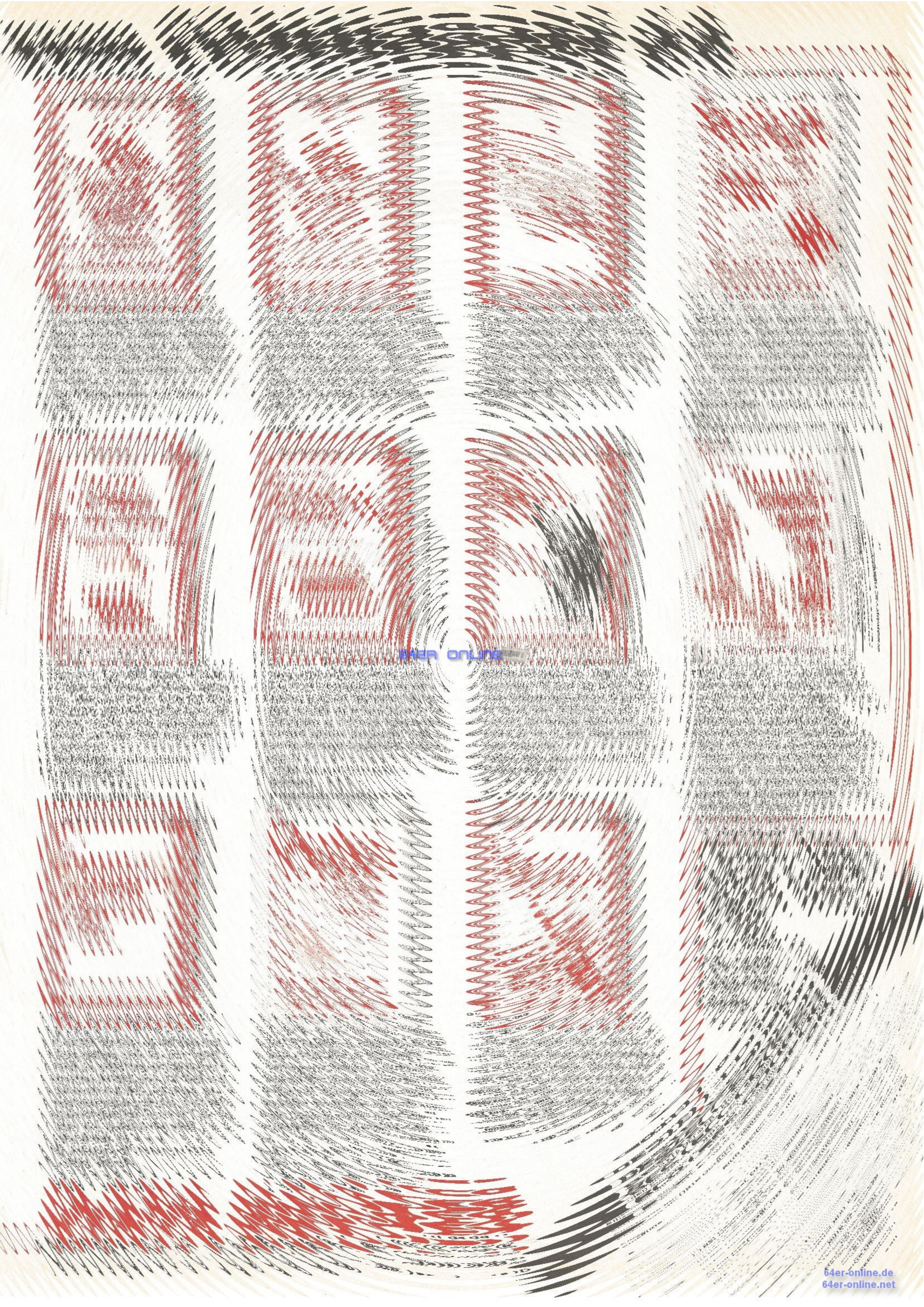
Bild 12. Textumwandlung

Auch hier (Listing 23) müssen wir wieder die Bedingungen mit NOT negieren, damit die normale Logik im Programmfluß erhalten bleibt.

Damit sind wir mit der Beschreibung der Steuerbausteine am Ende.

Allerdings nicht zu Ende sind wir mit dem Thema des strukturierten Programmierens. Wer sich mit Steuerbausteinen der beschriebenen Art begnügt und ansonsten wie üblich vor sich hin programmiert, der programmiert noch lange nicht strukturiert. Damit ist noch kein logischer und übersichtlicher Aufbau des Programms gewährleistet. Sicher, ohne Steuerbausteine ist strukturiertes Programmieren nicht möglich, aber die Benutzung von Steuerbausteinen ist nur der erste Schritt. Zu diesen müssen noch zwei weitere Bausteintypen kommen: Unterprogramm- und Modulbausteine. Fortsetzung auf S. 154





Memory Map mit Wandervorschlägen (14)

Heute geht es um die Speicherzellen 256 bis 645. Sie enthalten den Stapelspeicher und neben vielen nützlichen Adressen den Tastaturpuffer.

Der Speicherbereich von 256 bis 511 wird von den meisten Home-Computern zur Speicherung von Adressen und Daten bei Programm-Unterbrechungen und -Verzweigungen verwendet. Warum gerade dieser Bereich dafür genommen wird, und wie dieser Speicher funktioniert, können Sie dem Texteinschub »Der Stapelspeicher« entnehmen.

Das von Commodore verwendete Basic benötigt allerdings dafür nur den Bereich von 319 bis 511, so daß die Adressen von 256 bis 318 für verschiedene andere Zwecke genutzt werden.

Adresse 256 bis 266 ($100 bis $10A)

Arbeitsspeicher für Umwandlung von Gleitkomma-Zahlen in ASCII-Werte, auch FAC (Fließkomma-Akku) genannt

Diese 11 Byte werden von einer Routine des Betriebssystems verwendet, um Werte zwischenzuspeichern, die bei der Umwandlung von Gleitkomma-Zahlen in ASCII-Werte oder in Werte der Funktion TI$ anfallen. Eine andere Routine verwendet den Bereich, um Zeichenketten (Strings) zu untersuchen.

Adresse 256 bis 318 ($100 bis $13E)

Arbeitsspeicher für Fehler bei der Eingabe vom Band

Alle Daten, die auf Band gespeichert werden, stehen dort doppelt in zwei identischen Blöcken hintereinander. Beim Laden in den Computer werden beide Blöcke miteinander verglichen, um Fehler zu finden und, wo möglich, sie zu korrigieren.

In diesem Bereich, der übrigens auch bei der Speicherzelle 256 anfängt, aber 63 Byte in Anspruch nimmt, werden beim Laden Angaben gespeichert, aus denen das Betriebssystem erkennen kann, welche Bytes fehlerhaft sind.

Adresse 319 bis 511 ($13F bis $1FF)

Stapelspeicher (Stack) des Mikroprozessors

Die Funktionsweise eines Stapelspeichers, auf englisch »stack«, ist im Texteinschub gleichen Namens erklärt.

Der Stapelspeicher hat prinzipiell die Aufgabe, bei allen Sprüngen oder Unterbrechungen innerhalb eines normalen Programmablaufs alle Adressen und Daten so zu speichern, daß am Ende der Unterbrechung das Programm wieder fortgesetzt werden kann.

Derartige Unterbrechungen und Sprünge treten in Basic bei den Befehlen GOSUB-RETURN und FOR-NEXT auf, genauso wie bei vielen Routinen des Betriebssystems. In Maschinensprache gibt es dafür sogar eigene Befehle. Heimo Ponnath hat sie alle in seinem Assemblerkurs (Folge 7 und 8 im 64'er, Ausgabe 2/85 und 3/85) sehr ausführlich beschrieben.

Da uns hier Basic mehr interessiert, gebe ich Ihnen nur kurz an, was im Stapel gespeichert wird, da der Stapelspeicher nur in Maschinensprache manipuliert werden kann.

Jeder FOR-TO-NEXT-Befehl belegt 18 Byte im Stapelspeicher.

Im ersten Byte steht als Kennung die Zahl 129. Byte 2 und 3 enthalten in Low/High-Byte-Darstellung einen Zeiger auf die Adresse, in der die durch das FOR definierte Schleifen-Variable (zum Beispiel K in FOR K=0 TO 3) gespeichert ist. Die nächsten 5 Byte sind für den Gleitkommawert von STEP reserviert, das Byte danach für das Vorzeichen von STEP. Danach folgt der Gleitkommawert von TO mit 5 Byte und in zwei weiteren Byte die Nummer derjenigen Zeile, auf die nach dem NEXT zurückgesprungen wird. In den letzten beiden Byte schließlich steht ein Zeiger auf der Adresse, in der das nächste Zeichen steht, welches nach Beendigung der FOR-TO-NEXT-Schleife gelesen werden muß.

Ein GOSUB-Befehl belegt 5 Byte im Stapelspeicher. Byte 1 enthält die Kennzahl 141. Ihr folgen zwei Byte für die Nummer der Zeile, auf die nach RETURN zurückgesprungen wird. Die letzten beiden Byte enthalten wieder einen Zeiger auf die Adresse, in der das nächste Zeichen steht, mit dem nach RETURN das Programm fortgesetzt wird.

Der Basic-Befehl DEF zur freien Definition von Funktionen belegt ebenfalls 5 Byte im Stapelspeicher. Ihre Verteilung ist dieselbe wie von GOSUB, mit dem einzigen Unterschied, daß statt der ersten Kennzahl irgendein anderer Wert verwendet wird, der aber keine Bedeutung hat.

Wenn so viele FOR...NEXT-Schleifen oder GOSUB-Sprünge gleichzeitig im Programm vorkommen, daß der Stapelspeicher voll wird, steigt das Programm mit OUT OF MEMORY aus.

Adresse 512 bis 600 ($200 bis $258)

Eingabespeicher von Basic

Wenn Sie Zeichen, zum Beispiel einen Befehl oder eine Programmzeile, eingeben und mit der RETURN-Taste abschließen, werden diese Zeichen in diesen Speicherbereich von 512 bis 600 gebracht. Seine Länge von 89 Byte entspricht der Länge einer logischen Zeile des VC 20 (88 Zeichen) plus einer Abschluß-Null. Die logische Zeilenlänge des C 64 von 80 Zeichen füllt den Speicherbereich nicht ganz aus, aber das Betriebssystem des C 64 ist nicht geändert worden.

Nach RETURN sucht der Computer diesen Eingabespeicher nach Gänsefüße, Komma und nach der Zahl für Zeilenende ab. Dann wandelt der Computer die gespeicherten Zeichen in für ihn lesbare Zahlen (Token und ASCII-Werte) um und fügt am Anhang die Zeilennummer und die Anschluß-Adresse (Link) der nächsten Zeile, am Ende die Abschluß-Null hinzu. Wenn eine Zeilennummer vorhanden ist, kommt alles in den Programmspeicher. Fehlt sie jedoch, dann wird die ganze Anweisung sofort ausgeführt (Direktmodus).

Eine detaillierte Beschreibung dieses Eingabe- und Umwandlungsvorganges gab Christoph Sauer im 3. Teil seines Kurses »Der gläserne VC 20« im 64'er, Ausgabe 11/84 ab Seite 126.

Dieser Speicherbereich wird auch von den Befehlen INPUT und GET benutzt, um die Eingabedaten aufzunehmen. Das erklärt übrigens, warum diese beiden Befehle nur innerhalb einer Programmzeile und nicht im Direktmodus verwendet werden können. Sie verwenden ja denselben Speicherplatz, der vom Direkt-(Eingabe-)Modus verwendet wird.

Es erklärt außerdem, warum eine von INPUT geforderte Eingabe maximal 88 Zeichen lang sein darf.

Adresse 601 bis 630 ($259 bis $276)

Tabellen für File-Nummern, Geräte-Nummern und Sekundär-Adressen von eröffneten Dateien

Bei der Besprechung der Speicherzelle 152 im 64'er, Ausgabe 6/85, Seite 122/123, habe ich diesen Speicherbereich bereits erwähnt. Ich habe damals gesagt und gezeigt, daß die Zelle 152 über die Anzahl der eröffneten Dateien (Files) Buch führt, die Tabellen in 601 bis 630 dagegen darüber, welche File-Nummern, Geräte-Nummern und Sekundär-Adressen jeder eröffneten Datei zugeordnet ist. Wer sich nochmals über diese Begriffe orientieren will, den verweise ich auf Teil 11 dieses Kurses (64'er, Ausgabe 10/85, Seite 134/135).

Der Speicherbereich von 601 bis 630 ist in drei Blöcke unterteilt.

In jeder Tabelle können also maximal 10 Byte stehen. Sie haben folgende Zusammensetzung:

601 bis 610	Tabelle der File-Nummern
611 bis 620	Tabelle der Geräte-Nummern
621 bis 630	Tabelle der Sekundär-Adressen

Die drei Angaben über eine eröffnete Datei stehen in den Tabellen jeweils am gleichen Platz. Wenn also die Datei Nummer 5 als dritte Datei eröffnet worden ist, steht eine 5 in Zelle 603, ihre Gerätenummer in Zelle 613, die Sekundär-Adresse entsprechend in Zelle 623.

Immer wenn eine neue Datei eröffnet wird, kommen diese Angaben auf die nächsten Plätze der Tabellen und der Inhalt der Speicherzelle 152 wird um 1 erhöht. Wird dagegen eine Datei geschlossen, dann rücken alle Angaben dahinter um eine Stelle zurück.

In diesen Tabellen kann nachgesehen werden, mit welchen Parametern Dateien eröffnet worden sind. Eine sehr interessante Anwendung, die den Vorschlag der beiden Autoren hier zitieren, wurde veröffentlicht [2].

Eine Änderung der Filenummern in Tabelle 601 bis 610 ist nicht empfehlenswert, um Verwechslungen zu vermeiden.

Mit dem folgenden kleinen Programm, welches bei einer eröffneten Datei die Geräte-Nummer in Tabelle 611 bis 620 ändert, kann zwischen einem Drucker mit Geräte-Nummer 4 und einem Plotter mit Geräte-Nummer 6 umgeschaltet werden.

```
10  OPEN 4,4,0
20  POKE 611,6:  PRINT#4, "PLOTTER"
30  POKE 611,4:  PRINT#4, "DRUCKER"
```

Ähnliches ist mit den Sekundär-Adressen möglich. Dabei muß man allerdings wissen, daß die Sekundär-Adressen nicht so wie sie sind in der Tabelle 621 bis 630 gespeichert werden, sondern mit »OR 96« verknüpft. Dasselbe müssen wir auch machen:

```
10  OPEN  3,4,0
20  POKE  621,0   OR 96: PRINT#3, "GRAFIKMODUS"
30  POKE  621,7   OR 96: PRINT#3, "TEXTMODUS"
```

Mit dieser Methode können Sie sich das Öffnen und Schließen vieler Dateien ersparen.

Adresse 631 bis 640 ($277 bis $280)

Tastaturpuffer

Bei der Behandlung der Speicherzelle 203 in Ausgabe 11/85 habe ich die Codezahlen beschrieben, die bei Drücken einer der 64 Tasten erzeugt werden. In der letzten Ausgabe (12/85) bei den Speicherzellen 245/246 haben wir gesehen, wie aus diesen Tastencodes der ASCII-Code für die verschiedenen Zeichen einer Taste umgerechnet wird.

Hier nun im Tastaturpuffer landen diese umgerechneten ASCII-Werte. Wenn Sie den Kurs schon länger verfolgen, ist Ihnen das auch nicht neu, denn in Ausgabe 7/85 habe ich die Wirkungsweise des Tastaturpuffers bei der »Dynamischen Tastenabfrage« sozusagen im Vorgriff, ausführlich erklärt.

Zur Erinnerung sei gesagt, daß im Tastaturpuffer alle Zeichen zwischengespeichert werden, die während eines Programmlaufes eingegeben und nicht sofort vom Betriebssystem verarbeitet werden können. Sobald der Computer sich im Eingabe-Modus befindet — nach Programmende oder bei INPUT- und GET-Befehlen, werden die Zeichen in der Reihenfolge ihrer Eingabe herausgeholt und verwendet.

Der Tastaturpuffer ist 10 Byte lang. In Speicherzelle 198 steht, wieviele Zeichen sich im Puffer befinden.

Als Ergänzung zu den Beispielen der dynamischen Tastenabfrage im 64'er, Ausgabe 7/85 zeige ich Ihnen im Texteinschub »Programme, die sich selbst verändern« noch ein paar andere Anwendungen.

Adresse 641 bis 642 ($281 bis $282)

Zeiger auf den Anfang des Programmspeichers

Wenn der Computer eingeschaltet wird oder wenn mit einer Reset-Taste beziehungsweise mit SYS58260 (VC 20: SYS 58232) ein Kaltstart ausgelöst wird, setzt das Betriebssystem diesen Zeiger auf die Adresse des ersten freien RAM-Speicherplatzes.

Beim C 64 ist dies die Adresse 2048. Beim VC 20 hängt sie von der Speichererweiterung ab; ohne Erweiterung ist es 4096, mit einer 3-KByte-Erweiterung dagegen 1024, mit 8 KByte oder mehr ist die Adresse 4608.

Dieser Zeiger wird vom Basic-Übersetzer in die Speicherzelle 43 übernommen und nur von dort weiter verwendet.

Adresse 643 bis 644 ($282 bis $283)

Zeiger auf das Ende des Programmspeichers

Dieser Zeiger ist der Zwilling zu dem anderen Zeiger in 641/642. Er wird vom Betriebssystem auf die Adresse gesetzt, welche beim Kaltstart beziehungsweise der dabei durchgeführten Prüfung des Speichers den letzten verfügbaren RAM-Speicherplatz angibt. Beim C 64 ist diese Adresse normalerweise 40960 ($A000), beim VC 20 ohne Erweiterung 7680.

Dieser Zeiger wird vom Basic-Übersetzer in die Speicherzelle 55 übernommen.

Adresse 645 ($285)

Flagge für Ein- und Ausschalten der IEEE-488-Karte

Diese Speicherzelle ist etwas mysteriös. Sie kommt im ganzen Betriebssystem nur ein einziges Mal zum Einsatz und zwar als Flagge beim Betrieb der sogenannten IEEE-488-Interface-Karte. Wenn diese Flagge gesetzt ist, wartet der Computer 64 Millisekunden lang, ob er von einem angeschlossenen Gerät angesprochen wird. Wenn kein Signal kommt, gibt er ein Fehlersignal aus.

Zahlen in der Zelle 645, die kleiner als 128 sind, bedeuten Flagge gesetzt, größer als 128 löschen sie die Flagge.

Das nächste Mal machen wir mit mehreren interessanten Speicherzellen für Schriftfarbe und Tastatursteuerung weiter.

(Dr. H. Hauck/ah)

Literatur

[1] Sheldon Leemon
Mapping the C 64
COMPUTE! Publications 1984
[2] Hannes Rügheimer, Christian Spanik
C 64 PEEK + POKE Adreßbuch
CHIP-Special
[3] Russ Davies
Mapping the VIC
COMPUTE! Publications 1984

Texteinschub #1: Der Stapelspeicher

Der normale Speicher, mit dem wir es immer zu tun haben, in den wir Zahlen hineinPOKEn oder herausPEEKen,ist aufgebaut wie eine Häuserreihe, in der jedes Haus seine eigene Adresse hat. Wer etwas in ein bestimmtes Haus abliefern oder aus ihm abholen will, muß seine genaue Adresse kennen.

Dieses Speicherprinzip heißt RAM, das ist die Abkürzung für »Random Access Memory« oder auf deutsch »Speicher mit wahlfreiem Zugriff«.

Der Stapelspeicher funktioniert anders. Jetzt werden die Häuser der Reihe nach aufgesucht, wie mit einer Postwurfsendung oder durch die Müllabfuhr. Ein besserer Vergleich ist der Aktenstapel. Die erste Akte wird auf den Tisch gelegt, alle nachfolgenden kommen obendrauf.

Beim Aktenstapel sieht man eine weitere Eigenschaft. Wenn er nämlich abgearbeitet wird, kommt die Akte als erste an die Reihe, die als letzte auf den Stapel gelegt worden ist.

Dieses Speicherprinzip heißt »LIFO«; das ist eine Abkürzung aus dem englischen und heißt »Last In First Out«.

Der Benutzer eines Stapelspeichers braucht sich nicht mehr um Adressen zu kümmern. Er kennt nur noch den Platz, wo der Speicher Daten annimmt beziehungsweise abgibt. Was der Benutzer sich allerdings merken muß, ist die Reihenfolge seiner Ein- und Ausgaben.

Stapelspeicher werden von mehreren Programmiersprachen verwendet; am bekanntesten ist wohl FORTH, auch die HP-(Hewlett-Packard)-Taschenrechner arbeiten mit diesem Prinzip.

Unsere Computer verwenden das Prinzip des Stapelspeichers nur bei Programmschleifen, Unterprogramm-Sprüngen, ja überhaupt bei jeder Unterbrechung des normalen Programmablaufes. Das Problem dabei ist nämlich, sich alle Adressen und Angaben des Programms so zu merken, daß nach dem Ende der Unterbrechung das alte Programm lückenlos fortgesetzt werden kann.

Der Stapel belegt den Speicherbereich von 256 bis 511. Dieser Bereich unterscheidet sich in seinem Aufbau natürlich überhaupt nicht von den anderen Speicherbereichen. Es wäre auch viel zu aufwendig, alle Daten des Stapelspeichers bei jeder Ein- und Ausgabe rumzuschieben. Diese Aufgabe besorgt ein Register des Mikroprozessors, das »Stapelzeiger« genannt wird und das wie ein Zähler arbeitet.

Am Anfang steht in diesem Stapelzeiger die oberste Adresse 511. Bei jeder Eingabe wird die Zahl um 1 erniedrigt, bei jeder Ausgabe um 1 erhöht. Da der Stapelzeiger, wie jedes andere Register auch, eine Länge von 8 Bit hat, kann er nur die Dualzahlen von 00000000 bis 11111111 darstellen. Um daraus 256 beziehungsweise 511 zu formen, stellt der Mikroprozessor sozusagen fest verdrahtet ein neuntes Bit, immer auf 1, der Zahl voran.

Texteinschub #2: Programme, die sich selbst verändern

Der Speicherbereich von 631 bis 640 beherbergt den Tastaturpuffer. Schon im 64'er, Ausgabe 7/85, Seite 141/142 habe ich Ihnen Anwendungen gezeigt, die den Tastaturpuffer einsetzen und in der Literatur unter dem Namen »Dynamische Tastenabfrage« bekannt sind.

Hinter diesem natürlich aus dem Englischen übersetzten Begriff verbirgt sich die Möglichkeit, innerhalb eines Programms bestimmte Werte in den Puffer zu speichern, die dort so lange aufgehoben bleiben, bis das Programm — aus welchen Gründen auch immer — beendet wird. Dann erst werden die Werte hervorgeholt. Wenn es Zeichen sind, dann erscheinen sie auf dem Bildschirm. Wenn es Steuertasten sind (zum Beispiel RETURN), werden sie ausgeführt.

Ich will hier nicht die in Ausgabe 7 gezeigten Anwendungen wiederholen, sondern sie lediglich um ein paar weitere Tricks ergänzen, die ich im Juli aus Platzgründen nicht bringen konnte.

Zeile in ein Programm einfügen

Die folgende Methode ermöglicht die Veränderung eines Programms durch sich selbst. Genauer gesagt, man kann damit Programmzeilen einfügen. Nehmen wir an, Sie haben folgendes Teilprogramm:

```
150  PRINT"BITTE TASTE DRUECKEN"
160  GET A$:IF A$=""THEN 160
170  IF A$ <> "E" THEN 150
180  END
...
...
500  PRINT "NAECHSTER TEIL'
```

Die ersten vier Zeilen warten so lange, bis die E-Taste gedrückt wird, dann bleibt das Programm mit READY stehen.

Wir wollen nun eine zusätzliche Zeile 165 durch das Programm einfügen lassen, mit der nicht beendet, sondern durch Drücken der Q-Taste auf die Zeile 500 gesprungen wird. Das erreichen wir durch folgende Zeilen:


```
172  PRINT CHR$(147)
174  PRINT "165 IF A$=CHR$(81) THEN 500"
176  PRINT "GOTO 150"
178  POKE 631,19:POKE 632,13:POKE 633,13:
     POKE 634,13:POKE 198,4
```

Wenn jetzt die E-Taste gedrückt wird, löscht Zeile 172 den Bildschirm, Zeile 174 druckt in die zweite Bildschirmzeile die neue Programmzeile 165 und Zeile 176 darunter ohne Zeilennummer den Direktbefehl GOTO 150. In den Tastaturpuffer werden nacheinander die Werte für HOME und dreimal RETURN eingegeben und die Anzahl der Zeichen im Puffer auf 4 begrenzt.

Sobald nun das Programm den Befehl END in Zeile 180 erreicht, werden die Werte im Puffer ausgeführt, das heißt der Cursor geht nach HOME, das erste RETURN setzt ihn ohne Wirkung eine Zeile tiefer, wo er auf der neu ausgedruckten Zeile 165 steht. Das zweite RETURN gibt diese Zeile ein, das dritte RETURN führt den Direktbefehl GOTO 150 aus, wodurch das Programm weiterläuft, so wie vorher. Der einzige Unterschied ist nur, daß jetzt eine neue Zeile im Programm steht, nämlich die Zeile 165.

Prüfen Sie es mit STOP und LIST.

Zeile verändern oder löschen

Diese Methode habe ich von S. Leemon [1] übernommen. Sie funktioniert im Prinzip genauso wie das Einfügen von Zeilen. Die Programmzeilen sind durch REMs erklärt. Probieren Sie es aus.

```
10  REM – DIESE ZEILE WIRD GELÖSCHT
20  PRINT CHR$(147):PRINT:PRINT
30  PRINT "60 LIST"
40  PRINT "10"
50  PRINT "GOTO 70" CHR$(19)
60  FOR I=631 TO 633:POKE I,13:NEXT: POKE 198,3:END
70  REM – DIESE ZEILE WIRD ERSETZT
```

Diese Verfahren, Zeilen eines Programms während des Laufs eines Programms zu ändern, haben sicher für einfache Programme keine große Bedeutung. Ich bin aber fest davon überzeugt, daß sie bei Programmen, die selbst lernen können, eine wichtige Rolle spielen. Nur habe ich hier jetzt noch kein gutes Demonstrationsbeispiel.

Einfügen einer Funktion

Sie alle kennen die Möglichkeit, Funktionen mit dem Basic-Befehl DEF FN.. selbst zu definieren. Vielleicht haben Sie auch schon einmal frustriert die Versuche abgebrochen, eine Funktion während des Programmablaufs per INPUT eingeben zu können. Das funktioniert nämlich nicht.

Mit dem Tastaturpuffer aber geht es, und zwar sehr elegant. Der vergebliche Versuch sieht so aus:

```
150  PRINT "FORMEL EINGEBEN"
160  INPUT "F(X)=";A$
...
250  DEF FNA(X)=A$
260  PRINT FNA(9)
270  GOTO 150
```

Der STRING A$, in Zeile 160 eingegeben, wird von Zeile 250 nicht übernommen. Zeile 250 können Sie so nicht verwenden, also löschen Sie sie bitte.

Sie können diese Zeile aber vom Programm einfügen lassen:

```
170  PRINT CHR$(147):PRINT:PRINT
180  PRINT "250 DEF FNA(X)=";A$
190  PRINT "GOTO 240"
200  PRINT CHR$(19)
210  POKE 631,13: POKE 632,13: POKE 198,2: END
...
...
240  REM – FORMELBERECHNUNG –
```

Wenn Sie nicht genau nachvollziehen können, was da vorgeht, empfehle ich Ihnen, an das Ende der Zeile 200 noch ein END anzuhängen. Dann bleibt das Programm dort stehen, Sie sehen, wo der Cursor steht und können dann die Wirkung der Zeile 210 in Ruhe überlegen.

Eine weitere interessante Anwendung habe ich bei Russ Davies gefunden [3]. Sie ermöglicht, Routinen des Betriebssystems oder Basic-Übersetzers aus dem Speicher auszulesen und als DATA-Anweisungen in Ihr Basic-Programm einzufügen.

Einfügen von DATA-Zeilen

Die Aufgabe besteht darin, aus einem auszuwählenden Speicherbereich — dort, wo die in Frage kommende Routine sitzt — den Maschinencode herauszuPEEKen und mit der inzwischen bekannten Methode in eine DATA-Zeile einzufügen. Ich gehe schrittweise vor:

```
20  PRINT CHR$(147): PRINT: PRINT
30  PRINT "5555 DATA";
50  FOR A=0 TO 3
60  PRINT PEEK(A)",";:NEXT
70  PRINT CHR$(19)
80  POKE 631, 13: POKE 632,13: POKE 198,2: END
```

Mit Zeilen 20 und 30 wird der DATA-Zeile die Zeilennummer 5555 gegeben und in die 3. Bildschirmzeile geschrieben. Zeile 50 liest die Speicherzellen 0 bis 3 aus und druckt sie, mit einem Komma versehen, dahinter. Zeilen 70 und 80 geben diese Zeile in gewohnter Manier in das Programm ein.

Ganz richtig ist das noch nicht, da nach LIST die Zeile 5555 zuviele Zwischenräume hat:

```
5555 DATA 47 , 55 , 0 , 170 ,
```

Zur Korrektur müssen wir Zeile 60 erweitern:

```
60 PRINT MID$(STR$(PEEK(A)),2)",";:NEXT
```

STR$() wandelt den durch PEEK ausgelesenen Zahlenwert in einen String um. MID$(...,2) schneidet die Leerstellen weg.

Mr. Davies hat daraus eine komfortable kleine Routine gebaut, die den auszulesenden Speicherbereich und die gewünschte Zeilennummer der ersten DATA-Zeile abfragt, dafür sorgt, daß eine DATA-Zeile nicht länger als 16 Zeichen wird, in diesem Fall die nächste Zeilennummer um 10 erhöht und sogar das letzte Komma in jeder DATA-Zeile eliminiert.

Dieser ganze Komfort hat natürlich nichts mit dem Tastaturpuffer zu tun. Dieser Teil bleibt unverändert. Aber ich finde dieses Programm so durchdacht, daß ich es Ihnen als ein Beispiel guter Programmierung angeben will.

```
10  INPUT "STARTADRESE";A: INPUT "ENDADRESE";E:
    INPUT "ZEILE";Z
20  PRINT CHR$(147)
30  PRINT Z "DATA";
40  IF A> E THEN END
50  FOR A=A TO A+15 + (E<A+15)*(A+15-E)
60  PRINT MID$(STR$(PEEK(A)),2)",";:NEXT
70  PRINT"{CRSR-links}":PRINT "A="A":
    E="E": Z="Z+10": GOTO 20";
80  POKE 631, 19: POKE 632,13: POKE 633,13:
    POKE 634,13: POKE 198,4: END
```

Streifzüge durch die Grafik-Welt (Teil 3)

C 128-Besitzer aufgepaßt: Eine Teilnahme an den Streifzügen ist auch für Sie möglich. Durch die »Übersetzung« des Standards starten auch Sie mit uns in der nullten Dimension.

Viele von Ihnen werden das neue Flaggschiff von Commodore, den C 128, schon vor sich stehen haben. Damit Sie nicht immer bei der Umsetzung der Beispielprogramme in den C 64-Modus schalten müssen, um HIRES-3 zu verwenden, soll hier die Übersetzung der von uns verwendeten, allgemeinen Grafikbefehle in das Basic 7.0 dieses Computers besprochen werden.

Der C 128 und unser Standard

Der erste Befehl (»INIT«), der die Initialisierung der Grafik (Bitmap einrichten, löschen und Farbgebung) bewirken soll, ist so umzusetzen:
COLOR 0,16 : COLOR 1,7 :
COLOR 4,1 : GRAPHIC 1,1

Komplizierter wird es da leider beim »START«-Befehl. Dieser soll das Grafiksystem in einen definierten Ausgangszustand bringen. Der Ursprung des Bildschirmsystems soll nach »START« in der Ecke links unten liegen und in der Horizontalen müssen 320, in der Vertikalen 200 Bildkoordinaten zur Verfügung stehen. Leider ist auch der Grafikbildschirm des C 128 so eingerichtet, daß von links oben an gezählt wird und die Y-Achse nach unten zeigt. Der »SCALE«-Befehl im Basic 7.0 erlaubt zwar die Skalierung der beiden Achsen bis zu Maximalwerten von 32767 (im Gegensatz zu den Angaben im Handbuch, wo maximal 1023 erwähnt sind). Es gibt aber keine Möglichkeit, das Koordinatensystem zu verschieben. Man muß sich also leider wieder mit einer Transformation begnügen, was — solange wir dazu nicht wieder ein Assemblerprogramm haben wie in HIRES-3 — natürlich Rechenzeit beansprucht.

In Listing 1 finden Sie ein kleines Unterprogramm, das — ausgehend von den schon aus HIRES-3 bekannten Systemgrenzwerten XU,XO,YU und YO (kleinste/größte X-, beziehungsweise Y-Koordinate) — aus den eingegebenen Koordinaten X und Y die Bildschirmkoordinaten XX und YY berechnet. Nun ist es relativ einfach, durch Festlegung dieser Systemgrenzwerte den Befehl »START« zu realisieren:
XU = 0 : XO = 319 : YU = 0 :
YO = 199

Ebenfalls mit Hilfe des Transformations-Unterprogrammes kann die Grafikanweisung »MITTE« übersetzt werden durch Definieren von:
XU = -160 : XO = 160 : YU = -100 :
YO = 100

Durch die Notwendigkeit der Transformation muß nun jede weitere Punktkoordinate vor dem jeweiligen Grafikbefehl durch unser Unterprogramm laufen. Der Grafik-Befehl »PUNKT (X,Y)« lautet dann:
GOSUB 1020 : DRAW 1,XX,YY

Für »LINIE (XA,YA,XB,YB)« steht beim C 128:
X = XA : Y = YA : GOSUB 1020 :
X1 = XX : Y1 = YY
X = XB : Y = YB : GOSUB 1020 :
X2 = XX : Y2 = YY
DRAW 1,X1,Y1 TO X2,Y2

Für »KREIS (XM,YM,RX,RY)« muß der Mittelpunkt anders transformiert werden als die Halbmesser:
X = XM : Y = YM : GOSUB 1020 :
X5 = XX : Y5 = YY
RX = RX*319/(XO-XU) :
RY = RY*199/(YO-YU)
CIRCLE 1,X5,Y5,RX,RY

Die weiteren Übersetzungen:
TEXT (A$,XT,YT) entspricht CHAR 1,XT,YT,A$
SHOW entspricht GRAPHIC1
NORMAL entspricht GRAPHIC0
GRESET entspricht SCNCLR(1)

Damit ergeben sich die C 128-Versionen der beiden Programme aus der letzten Folge (Fensterrose und Spiralen), die hier als Listing 2 und Listing 3 abgedruckt sind.

In Tabelle 1 finden Sie eine Übersicht über alle Grafikbefehle und ihre Übersetzungen in die drei Systeme: C 64 + HIRES-3, Plotter 1520, C 128 Basic 7.0.

```
1000 REM *** UP TRANSFORMATION ********
1001 REM EINMAL ZU DEFINIEREN SIND:
1002 REM  XU,XO = NIEDRIGSTER,HOECHSTER
1003 REM              X-WERT
1004 REM  YU,YO = NIEDRIGSTER,HOECHSTER
1005 REM              Y-WERT
1006 REM FUER JEDEN PUNKT ZU UEBERGEBEN
1007 REM  X,Y = PUNKTKOORDINATEN AUS
1008 REM            BELIEBIGEM CARTESISCHEM
1009 REM            KOORDINATENSYSTEM
1010 REM AUSGEGEBEN WERDEN DANN:
1011 REM  XX,YY = BILDSCHIRMKOORDINATEN
1012 REM              DES 320/200-SYSTEMS
1013 REM ******************************
1020 XX=319*(X-XU)/(XO-XU)
1030 YY=199*(YO-Y)/(YO-YU)
1040 RETURN

READY.
```

Listing 1. C 128: Das Unterprogramm Transformation

Grafikstandard-probleme

Nicht daß Sie denken, der Sinn für Proportionen sei uns abhanden gekommen, weil sich dieses Kapitel mit verschiedenen Grafikstandards (kurz GKS genannt) in bezug auf unsere Computer befaßt: Es scheint mir aber doch notwendig, zumindest einmal kurz das Verhältnis beispielsweise des GKS und ähnlicher Normen des hier verwendeten Minimal-Grafik-Sprachschatzes anzureißen.

```
1 REM ********************************
2 REM *                              *
3 REM *          FENSTERROSE         *
4 REM *    VERSION FUER DEN PC 128   *
5 REM *                              *
6 REM *  HEIMO PONNATH  HAMBURG 1985 *
7 REM *                              *
8 REM ********************************
9 REM ********* EINGABEN ***********
10 COLOR0,1:COLOR4,1:COLOR5,6:SCNCLR
15 INPUT"ANZAHL DER STUETZPUNKTE";N
20 INPUT"RADIUS";R:INPUT"TEXTORT XT,YT";
XT,YT
25 REM ******** INIT **************
30 COLOR0,16:COLOR1,7:COLOR4,1:GRAPHIC1,
1
35 REM ******** START/MITTE *******
40 XU=-160:XO=160:YU=-100:YO=100
55 REM ***************************
60 D=2*π/N:DIMX(N),Y(N)
65 REM ******** KREIS **************
70 X=0:Y=0:GOSUB1020:X5=XX:Y5=YY:RX=R*31
9/(XO-XU):RY=R*199/(YO-YU)
72 CIRCLE1,X5,Y5,RX,RY
75 REM ***************************
80 GOSUB200
90 A$="FENSTERROSE"
95 REM ******** TEXT **************
100 CHAR1,XT,YT,A$
105 REM ******* SHOW(HIER UNNOETIG)
115 REM **************************
120 GETKEYB$
125 REM ******* NORMAL ************
130 GRAPHIC0:COLOR0,1
135 REM ******* GRESET ************
140 SCNCLR(1)
160 END
200 REM ******* UP ZEICHNEN *******
210 FORI=1TON:T=T+D:X(I)=R*COS(T):Y(I)=R
*SIN(T):NEXTI
220 S=N-1
230 FORI=1TOS:Z=I+1
240 FORJ=ZTON
242 REM ******* LINIE *************
243 X=X(I):Y=Y(I):GOSUB1020:X1=XX:Y1=YY:
X=X(J):Y=Y(J):GOSUB1020:X2=XX:Y2=YY
245 DRAW1,X1,Y1TOX2,Y2
247 REM **************************
250 NEXTJ:NEXTI:RETURN
1000 REM *** UP TRANSFORMATION ********
1020 XX=319*(X-XU)/(XO-XU)
1030 YY=199*(YO-Y)/(YO-YU)
1040 RETURN

READY.
```

Listing 2. Fensterrose C 128-Version



Bemühungen, eine geräteunabhängige Grafikschnittstelle softwaremäßig zu realisieren, gibt es schon geraume Zeit. In den USA existiert das System CORE, in Deutschland wurde vor einiger Zeit ein GKS als DIN-Entwurf vorgelegt. Beide sind von Haus aus für Großrechenanlagen entwickelt worden und wurden von dem rasanten technischen Fortschritt überholt, der Mikro-, Personal- und sogar Homecomputer im Feld der professionellen Grafikprogrammierung auftauchen ließ. IBM versucht, für den IBM/AT eine GKS-Implementierung zu erstellen. Die erforderlichen Rechenkapazitäten für GKS und auch CORE sind aber relativ hoch, weshalb eine Verwendung dieser Standards bislang nur auf leistungsfähigen Großcomputern oder 32-Bit-Geräten mit arithmetischen Coprozessoren sinnvoll ist. Einem 8-Bit-Computer sind Grenzen gesetzt, die es allenfalls erlauben, eine minimale Teilmenge des Standardwortschatzes anzustreben.

Die Frage ist, ob das wünschenswert erscheint. Dazu einige Erklärungen: Man kann grob zwei grundlegende Arten von Grafik-Befehlen unterscheiden, nämlich einmal mit und einmal ohne einen Grafik-Cursor. Im ersten Fall gleitet der »Zeichenstift« immer von der aktuellen Position des — häufig nur gedachten — Cursors zeichnend oder nicht zeichnend zur angegebenen neuen Position. Im anderen Fall existiert solch eine Cursorposition nicht. Alle Größen, die zur Ausführung eines Befehls wie LINIE nötig sind, werden mit diesem als Argument übergeben.

Im CORE-System (auch im GKS) gibt es beispielsweise folgende Befehle:

LINE-ABS-2(X,Y):

Zieht eine Linie von der aktuellen Cursorposition bis X,Y. Die 2 kommt von 2D, also einem ebenen Koordinatensystem mit zwei Achsen.

LINE-REL-2(DX,DY):

Zieht von der Cursorposition aus eine Linie zu einem Punkt, dessen Koordinaten um DX und DY von der Cursorposition verschieden sind.

MOVE-ABS-2(X,Y):

Der Cursor wird nicht-zeichnend von der aktuellen Position zum Punkt X,Y bewegt.

MOVE-REL-2(DX,DY):

Wie beim entsprechenden LINE-Befehl, aber nicht zeichnend.

Auf dem C 128 können wir auch diese Art der Grafikprogrammierung durchführen. Die Entsprechungen wären (in der gleichen Reihenfolge):

DRAW1 TO X,Y
DRAW1 TO (RDOT(0)+DX),
(RDOT (1)+DY)
LOCATE X,Y
LOCATE (RDOT(0)+DX),
(RDOT (1)+DY)

Sehen wir uns den Unterschied am Beispiel in Bild 1 an, wo ein Haus auf den normalen Commodore-Bildschirm gezeichnet werden soll.

Dem CORE-System nachgebildet, würden die Befehle folgendermaßen lauten:

LOCATE 50,50
DRAW1 TO 50,80
DRAW1 TO 110,80
DRAW1 TO 110,50
DRAW1 ,30
DRAW1 TO 50,50

In unserer Grafiksprache müßten wir schreiben:

LINIE (50,50,50,80)
LINIE (50,80,110,80)
LINIE (110,80,110,50)
LINIE (110,50,80,30)
LINIE (80,30,50,50)

Schon in diesem Beispiel, wo die für das CORE-System günstigeren miteinander verbundenen Linien zu zeichnen sind, haben wir einen Befehl mehr zu verarbeiten. Ganz deutlich tritt der Unterschied zutage, wenn nicht verbundene Strecken zu zeichnen sind. Dann muß vor je-

```
1 REM **********************************
2 REM *                                *
3 REM *    VERSCHLUNGENE SPIRALEN      *
4 REM *    VERSION FUER DEN PC128      *
5 REM *                                *
6 REM *  HEIMO PONNATH  HAMBURG 1985    *
7 REM *                                *
8 REM **********************************
9 REM ******** EINGABEN ***********
10 COLOR0,1:COLOR4,1:COLOR5,6:SCNCLR
15 INPUT"ANZAHL SPIRALEN";N:INPUT"TEXTST
ART XT,YT";XT,YT
20 INPUT"RADIUS,STARTWINKEL";R,A
22 REM ******* INIT ***************
23 COLOR0,16:COLOR1,7:COLOR4,1:GRAPHIC1,
1
25 REM ******* START/MITTE *******
30 XU=-160:XO=160:YU=-100:YO=100
32 FAST:GOSUB100
33 REM ******* KREIS **************
35 X=0:Y=0:GOSUB1020:X5=XX:Y5=YY:RX=R*31
9/(XO-XU):RY=R*199/(YO-YU)
37 SLOW:CIRCLE1,X5,Y5,RX,RY
40 A$="VERSCHLUNGENE SPIRALEN"
42 REM ******* TEXT ***************
45 CHAR1,XT,YT,A$
50 REM ******* SHOW (HIER UNNOETIG) **
55 GETKEYB$
60 REM ******* NORMAL *************
62 GRAPHIC0:COLOR0,1
65 REM ******* GRESET *************
67 SCNCLR(1)
70 END
100 REM ****** UP ZEICHNEN *******
102 NN=N*100:RA=R/NN:DIMT(NN)
105 FORJ=1TON:A=A+2*π/N
110 T=A:TH=2*π/100:T(0)=T
115 FORI=1TONN:T(I)=T(I-1)+TH:RR=RA*I
117 X=(RR-RA)*COS(T(I-1)):Y=(RR-RA)*SIN(
T(I-1)):GOSUB1020:XA=XX:YA=YY
119 X=RR*COS(T(I)):Y=RR*SIN(T(I)):GOSUB1
020:XB=XX:YB=YY
120 DRAW1,XA,YATOXB,YB
125 NEXTI
130 NEXTJ
135 RETURN
1000 REM *** UP TRANSFORMATION ********
1020 XX=319*(X-XU)/(XO-XU)
1030 YY=199*(YO-Y)/(YO-YU)
1040 RETURN

READY.
```

Listing 3. Spiralen C 128-Version

Befehl	HIRES-3 auf C 64	Plotter 1520	C 128 Basic 7.0
INIT	POKE 53280,0: SYS37498: HFL,6,12	OPEN1,6,1:OPEN2,6,2: PRINT#2,0:CLOSE2	COLOR,16:COLOR1,7: COLOR4,1:GRAPHIC1,1
START	TRS,0,320,0,200	PRINT#1,"M",0,-200: PRINT#1,"J"	XU=0:XO=319:YU=0:YO=199 Bezogen auf das UP Transformation
MITTE	TRS,-160,160, -100,100	PRINT#1,"R",240,0: PRINT#1,"J"	XU=-160:XO=160:YU=-100: YO=100 Bezogen auf das UP Transformation
PUNKT (X,Y)	TPK,X,Y	PRINT#1,"R",X,Y:PRINT#1, "J",X+2,Y+2:PRINT#1,"R",X,Y	GOSUB 1020:DRAW1,XX,YY UP Transformation ab 1020
LINE (XA,YA,XB,YB)	TLN,XA,YA, XB,YB	PRINT#,"R",XA,YA: PRINT#1,"J",XB,YB	X=XA:Y=YA:GOSUB 1020:XF =XX:X1=YY:X=XB:Y=YB: GOSUB 1020:X2=XX:Y2=YY: DRAW1,X1,Y1 TO X2,Y2 UP Transformation ab 1020
KREIS (XM,YM,RX,RY)	TKR,XM,YM,RX, RY,2*π	PROGRAMM: M=50:D=360·π/(M·180): DIMT(M):T(O)=2·π: FORJ=1 TO M:T(J)=T(J-1)+D: PRINT#1,"R",RX·COS(T(J-1)) +XM,RY·SIN(T(J-1))+YM: PRINT#1,"J",RX·COS(T(J)) +XM,RY·SIN(T(J))+YM:NEXTJ	X=XM:Y=YM:GOSUB 1020:X5 =XX:Y5=YY:RX=·319/ (XO-XU):RY=RY·199/(YO-YU): CIRCLE1,X5,Y5,RX,RY UP Transformation ab 1020
TEXT (A$,XT,YT)	TEX,A$,YT,XT	OPEN4,6:PRINT#1,"R",XT,YT: PRINT#4,A$:CLOSE4: PRINT#1,"M",240,-YT: PRINT#1,"J"	CHAR1,XT,YT,A$
SHOW	HAN	PRINT#1,"R",O,-200	GRAPHIC1
NORMAL	HOF	CLOSE1	GRAPHIC0
GRESET	LOE:AUS	OPEN7,6,7:PRINT#7:CLOSE7	SCNCLR(1)

Tabelle 1. Alle Grafikbefehle und ihre Übersetzungen

den DRAW-Befehl ein LOCATE treten, das den Cursor auf den Startpunkt einer Linie setzt. Das kostet die auf einem 8-Bit-Computer arbeitenden Grafiksysteme wertvolle Zeit.

Nebenbei sei noch bemerkt, daß das Basic 7.0 seinem DRAW-Befehl ein Zwitterdasein verliehen hat, wie Sie aus dem Vergleich der Übersetzung für den LINIE-Befehl und den CORE-Befehlen ersehen konnten. Das obige Programmproblem läßt sich beim C 128 sogar in einem Befehl erledigen:
DRAW 1,50,50 TO 50,80 TO 110,80 TO 110,50 TO 80,30 TO 50,50

Aus Gründen der Rechenzeiten, der besseren Überschaubarkeit eines Befehles und weil es einfach zu kompliziert wäre, HIRES-3 cursororientiert zu gestalten, wurde die cursorfreie Methode für unsere allgemeinen Grafikbefehle gewählt.

Noch mal etwas fürs Auge

Weil die Anwender des C 128 in den bisher vorgestellten gerätespezifischen Programmen zu kurz gekommen sind, sei ihnen als Trost das Listing 3 angeboten. Es erzeugt interessante Effekte durch Zeichnen ineinander verschachtelter, gegeneinander verdrehter Quadrate (siehe Bild 2).

Frei wählbar ist die Größe, der Drehwinkel, die Anzahl und der Ort auf dem Bildschirm. Bei geschickter Programmierung können die Ergebnisse als sogenannte Shapes miteinander kombiniert werden.

Nulldimensional: der Punkt

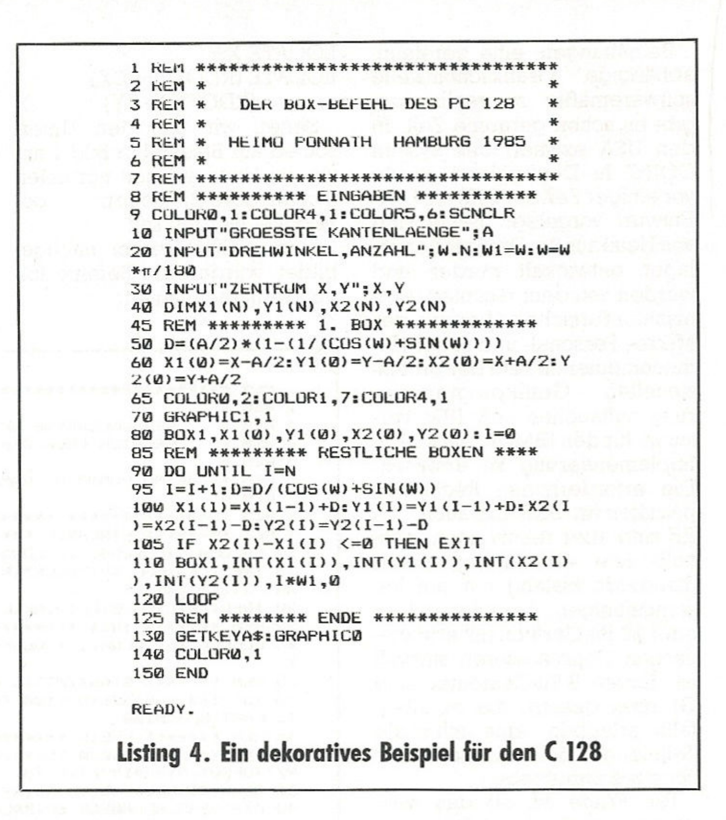
Betrachten Sie bitte alles bisher Behandelte als Rüstzeug, das wir für unsere Streifzüge durch die Grafikwelt nunmehr angelegt haben. Unser Weg wird uns durch alle Dimensionen führen: Vom Punkt zur 4.Dimension, um diesen Titel vom berühmten Colerus zu entleihen. Und damit haben wir sie auch schon angesprochen: die vielgehaßte, aber auch vielgeliebte Mathematik. Denn professionelle Computergrafik ohne dieses Instrument ist nicht denkbar. Sie, die Sie Ihre letzte Begegnung mit dieser Wissenschaft irgendwann einmal in der Schule hatten, verzagen Sie bitte nicht! Was wir an Voraussetzungen brauchen, haben Sie in der 10.Klasse gelernt: Einfache Gleichungen mit einer oder mehreren Unbekannten zu lösen. Alles andere wird erklärt werden und das ohne trockenen, mathematischen Jargon.

Nach diesen Vorreden soll gleich der erste Begriff erklärt werden: die Dimension. Das Wort stammt aus dem Lateinischen und bedeutet Ausdehnung. So hat eine ebene Fläche davon zwei: Länge und Breite. Deshalb spricht man hier von zweidimensionalen Gebilden. Eine Linie hat nur ihre Länge als Ausdehnung, sie ist also eindimensional. Fassen wir zusammen:

Dimension 0:	Punkt	———
Dimension 1:	Linie	Länge
Dimension 2:	Fläche	Länge Breite
Dimension 3:	Körper	Länge Breite, Höhe
Dimension 4:	???	Länge Breite, Höhe, ???

In noch höhere Sphären werden wir uns nicht versteigen, obwohl manche Esoteriker der Ansicht sind, daß im Leben des Menschen sechs Dimensionen eine Rolle spielen.

Beginnen wir also mit einem Gebilde der nullten Dimension, dem Punkt. Nun ist der Punkt an sich sicherlich ein wichtiges Thema für Philosophen, uns als Computergrafiker interessiert er aber immer in Verbindung irgendeiner Ortsbestimmung. Und weil wir — vor dem Bild-



```
1 REM ***********************************
2 REM *                                 *
3 REM *    DER BOX-BEFEHL DES PC 128    *
4 REM *                                 *
5 REM *  HEIMO PONNATH  HAMBURG 1985    *
6 REM *                                 *
7 REM ***********************************
8 REM ********** EINGABEN ***********
9 COLOR0,1:COLOR4,1:COLOR5,6:SCNCLR
10 INPUT"GROESSTE KANTENLAENGE";A
20 INPUT"DREHWINKEL,ANZAHL";W.N:W1=W:W=W
*π/180
30 INPUT"ZENTRUM X,Y";X,Y
40 DIMX1(N),Y1(N),X2(N),Y2(N)
45 REM ********* 1. BOX *************
50 D=(A/2)*(1-(1/(COS(W)+SIN(W))))
60 X1(0)=X-A/2:Y1(0)=Y-A/2:X2(0)=X+A/2:Y
2(0)=Y+A/2
65 COLOR0,2:COLOR1,7:COLOR4,1
70 GRAPHIC1,1
80 BOX1,X1(0),Y1(0),X2(0),Y2(0):I=0
85 REM ********* RESTLICHE BOXEN ****
90 DO UNTIL I=N
95 I=I+1:D=D/(COS(W)+SIN(W))
100 X1(I)=X1(I-1)+D:Y1(I)=Y1(I-1)+D:X2(I
)=X2(I-1)-D:Y2(I)=Y2(I-1)-D
105 IF X2(I)-X1(I) <=0 THEN EXIT
110 BOX1,INT(X1(I)),INT(Y1(I)),INT(X2(I)
),INT(Y2(I)),I*W1,0
120 LOOP
125 REM ******** ENDE ***************
130 GETKEYA$:GRAPHIC0
140 COLOR0,1
150 END

READY.
```

Listing 4. Ein dekoratives Beispiel für den C 128

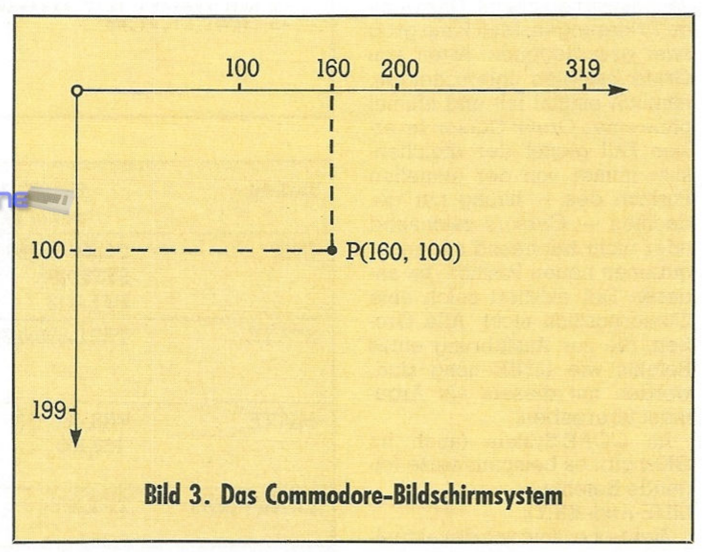


Bild 3. Das Commodore-Bildschirmsystem

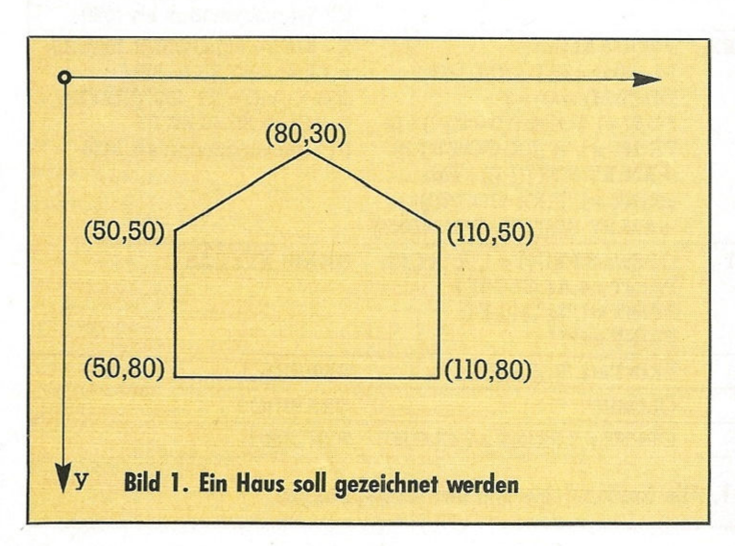


Bild 1. Ein Haus soll gezeichnet werden

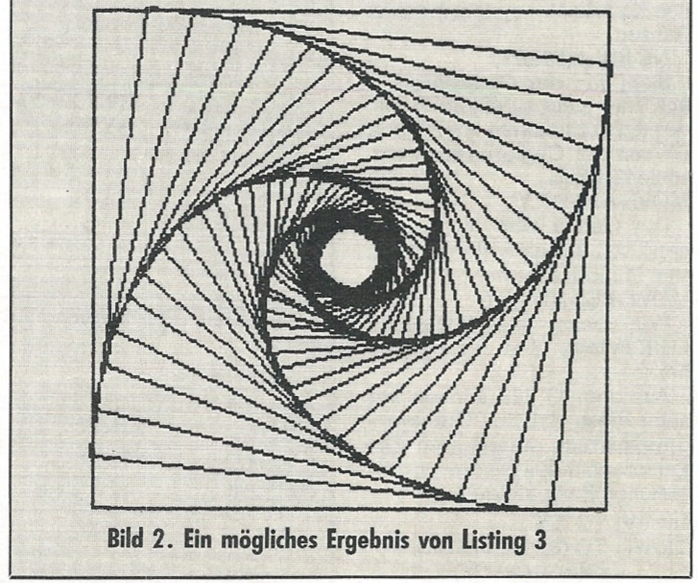


Bild 2. Ein mögliches Ergebnis von Listing 3





schirm sitzend — eine Fläche vor uns haben, wollen wir auf dieser Fläche auch den Punkt sehen. Wie bestimmen wir, daß der Punkt an einem ganz bestimmten Ort auftauchen soll? Wir unterteilen unseren Bildschirm — der Computerhersteller hat das für uns schon erledigt — in einen Raster von 320 waagerechten und 200 senkrechten Positionen. Diese numerieren wir: Die horizontalen Positionen von links anfangend nach rechts und die vertikalen oben beginnend nach unten. Was sich auf diese Weise ergibt, nennt man ein Koordinatensystem (vom lateinischen »coordinare«, was »zuordnen« bedeutet). Jeder Punkt kann nun durch Angabe zweier Koordinaten in diesen Raster gelegt werden. Allgemein nennt man die der Waagerechten zugeordneten Zahlen X-Koordinaten und die anderen Y-Koordinaten. Ein Punkt wird dann häufig in der Form P(X,Y) charakterisiert. Beispielsweise liegt der Punkt P(160,100) genau in der Bildschirmmitte. Die Koordinatenachsen sind Linien, die durch den Punkt P0(0,0) laufen. Die X-Achse als Linie gesehen, enthält auf ihrer gesamten Länge nur Punkte mit der Y-Koordinate 0, umgekehrt ist es mit der Y-Achse, die überall den X-Wert 0 aufweist. Was sich auf diese Weise ergibt, ist das Koordinatensystem, wie es in Bild 3 gezeigt ist.

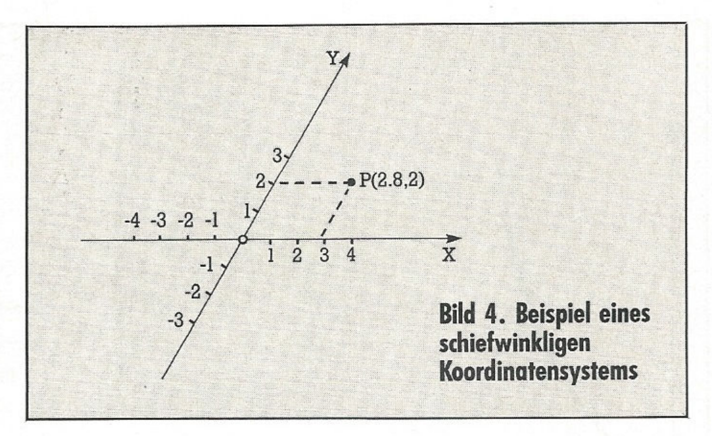


Bild 4. Beispiel eines schiefwinkligen Koordinatensystems

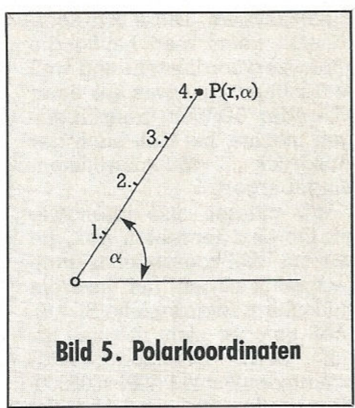


Bild 5. Polarkoordinaten

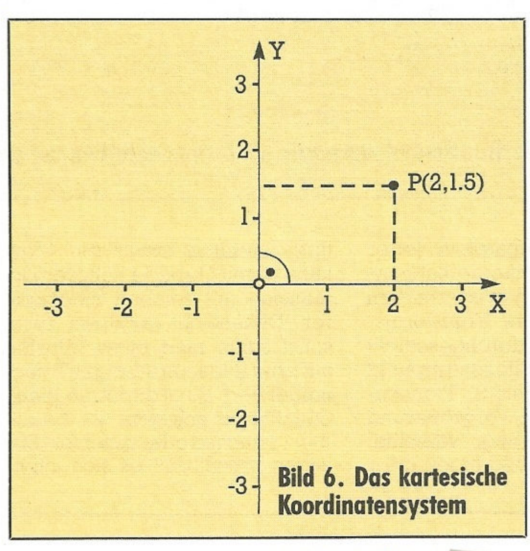


Bild 6. Das kartesische Koordinatensystem

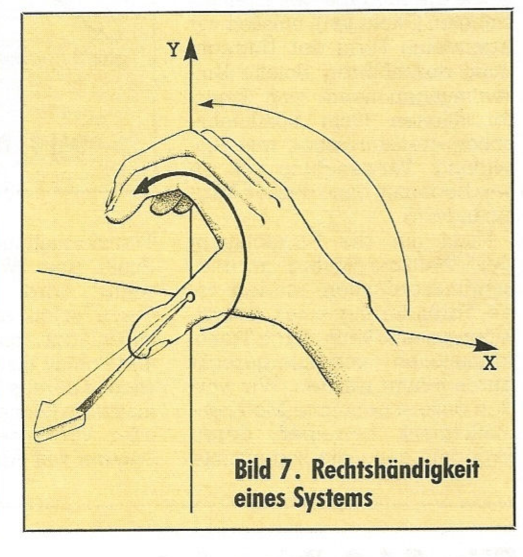


Bild 7. Rechtshändigkeit eines Systems

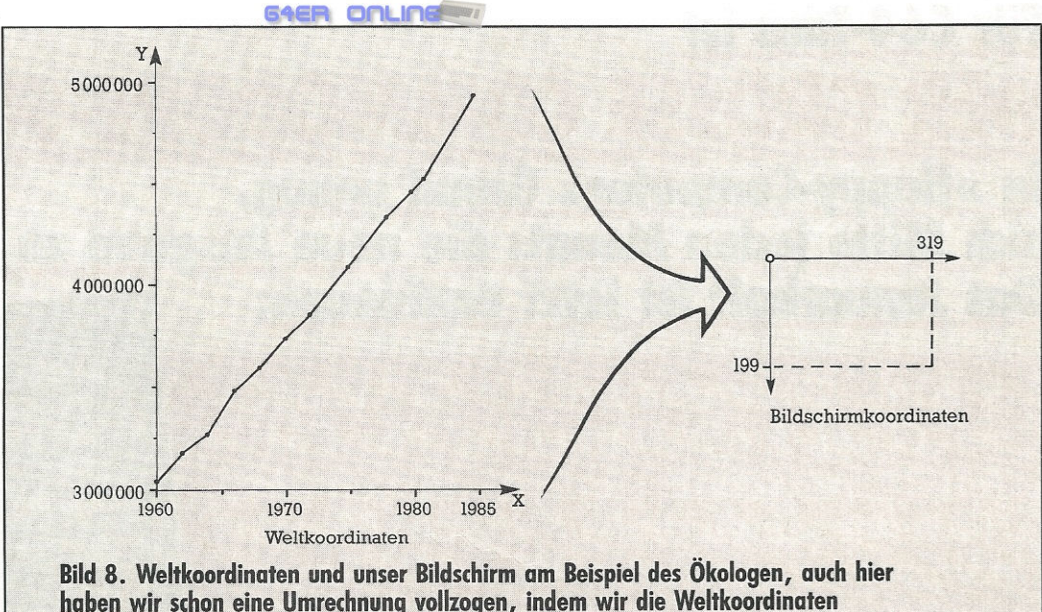


Bild 8. Weltkoordinaten und unser Bildschirm am Beispiel des Ökologen, auch hier haben wir schon eine Umrechnung vollzogen, indem wir die Weltkoordinaten in auf dem Papier darstellbare Längen übersetzt haben.

Koordinatensysteme

Den Ort von Punkten zu charakterisieren, ist im Prinzip noch auf vielerlei Weise möglich. So gibt es schiefwinklige Koordinatensysteme (siehe Bild 4) und manchmal verwendet man auch das sogenannte »polare« Koordinatensystem (siehe Bild 5). Hier werden die Länge des aus dem Ursprung kommenden Strahls und der Winkel, den dieser mit der Horizontalen bildet, zur Festlegung eines Ortes verwendet. Gebräuchlichstes Koordinatensystem allerdings ist das »kartesische«, angeblich benannt nach dem französischen Philosophen und Mathematiker Rene Descartes (von ihm stammt der berühmte Ausspruch »Ich denke, also bin ich«). Bild 6 zeigt ein kartesisches System, das durch die aufeinander senkrecht stehenden Achsen charakterisiert ist. Im allgemeinen verwendet man ein »rechtshändiges« System.

Transformationen

Rechtshängig deshalb, weil beim Drehen der X-Achse um den Ursprung (also um den Punkt P0(0,0)) diese auf dem kürzesten Weg zur Y-Achse die Richtung einschlägt, in die die Finger der rechten Hand zeigen. Der Daumen weist dabei auf den Betrachter (siehe Bild 7).

Nun erkennen Sie vielleicht schon, daß Commodore uns ein reichlich ungebräuchliches Koordinatensystem beschert hat: Es ist nämlich linkshändig aufgebaut. Außerdem ist die Skala sowohl auf der X-, als auch auf der Y-Achse sehr unpraktisch, denn Punkte wie zum Beispiel P1(-1,4) oder P2(3,-5) und P3(-12,-3) können nicht gezeichnet werden, und ein Versuch, das trotzdem zu tun, endet mit einer Fehlermeldung.

Weiterhin sollten wir uns überlegen, daß Computergrafik für die verschiedensten Zwecke gebraucht wird: Ein Architekt will den Grundriß eines Hauses zeichnen, das in Richtung Norden 15 Meter und in Richtung Osten 18 Meter Ausdehnung besitzt. Ein Geschäftsmann möchte im Rahmen einer Kosten-/Nutzen-Analyse ein Diagramm erstellen, das Arbeitsaufwand in Stunden gegen produzierte Menge eines Artikels aufzeigt. Ein Ökologe zeichnet ein Schaubild, welches die zahlenmäßige Entwicklung der Weltbevölkerung in den letzten 50 Jahren darstellen soll. Kurz: Jeder von Ihnen benötigt ein anderes Koor-

dinatensystem. Diese Art Koordinaten nennt man häufig die »Benutzerkoordinaten« und weil jeder Benutzer etwas aus einer 2D- oder 3D-Welt grafisch zeigen möchte, hat sich auch der Ausdruck »Weltkoordinaten« eingebürgert.

Wir müssen also irgendwie die Objekte der realen Welt, die wir im Weltkoordinatensystem festgelegt haben, auf unserem Bildschirm zeigen (siehe Bild 8). Weil es aber nicht möglich ist, auf dem Commodore-Bildschirm den Punkt P(1981,4508000) darzustellen (das ist beispielsweise die Aufgabe des oben erwähnten Ökologen), müssen wir irgendeine Form der Umrechnung durchführen. Solche Umrechnungen nennt man Transformationen (vom spätlateinischen »transformatio«, was Umbildung, Verwandlung von etwas bedeutet, ohne dessen Wert zu ändern).

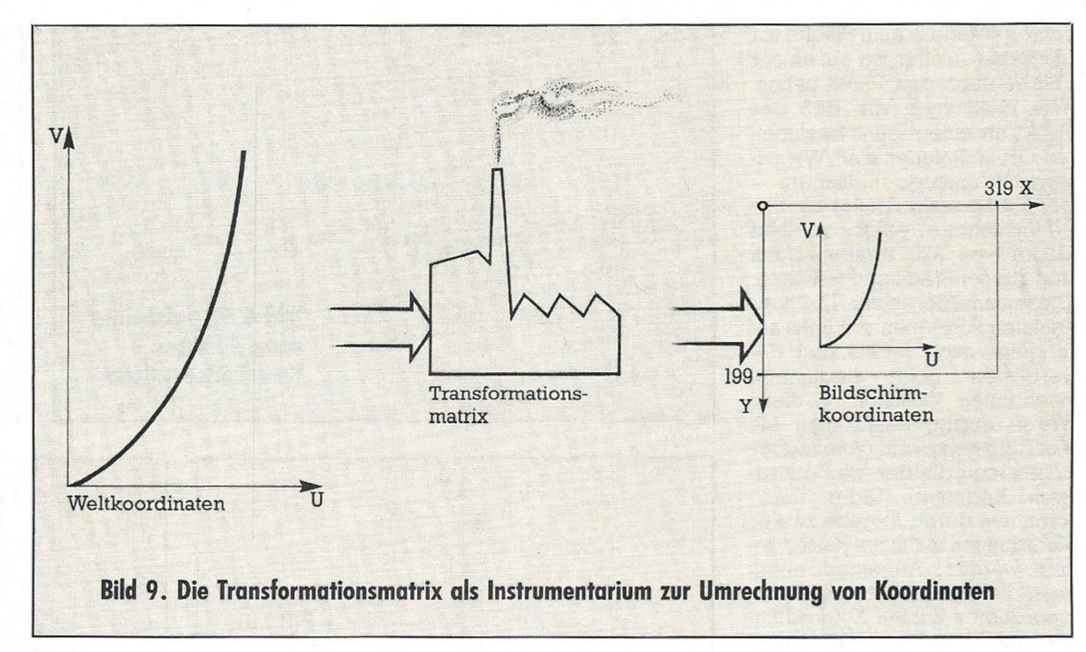


Bild 9. Die Transformationsmatrix als Instrumentarium zur Umrechnung von Koordinaten

Nicht nur die Umrechnung der Weltkoordinaten in Bildschirmkoordinaten, sondern jede Veränderung eines Koordinatensystems kann durch Transformationen computergerecht durchgeführt werden. Wir werden dazu sogenannte Matrizenmathematik betreiben, deren Ergebnis dann die sogenannte Transformationsmatrix ist. Jeder Punkt des Weltkoordinatensystems wird gewissermaßen durch solch eine Transformationsmatrix hindurchgeschickt und kommt als Bildschirmpunkt wieder heraus (Bild 9). Transformationen können Vergrößerung oder Verkleinerung, Verschiebungen von Abbildungen oder ihre Drehung bewirken. Man kann damit Gebilde höherer Dimension auf solchen niedrigerer Dimension abbilden (wie sonst sollte man einen Würfel auf dem Bildschirm zeigen?) und umgekehrt. Es ist damit möglich, Objekte zu spiegeln, zu krümmen, stauchen oder zu zerren. Sie sehen schon, daß es sich lohnt, den Umgang mit Transformationsmatrizen kennenzulernen.

Unser Thema in der nächsten Folge werden also Matrizen sein und wie man Computer dazu bringen kann, uns die Hauptarbeit mit diesen merkwürdigen mathematischen Gebilden abzunehmen.

(Heimo Ponnath/tr)

Von Basic zu Assembler Teil 1

Vom vertrauten Basic zum Programmieren in 6502-Assembler ist es gar nicht so weit. Wir werden langsam anfangen. Zuerst sind die Schleifen dran. Wir werden dabei Basic mit Maschinensprache vergleichen.

Man kann wohl getrost davon ausgehen, daß ein großer Teil der Commodore 64-Benutzer Basic beherrscht. Vermutlich hat es Sie aber auch schon öfters gereizt, die langsamen Programme durch Assembler-Routinen zu beschleunigen. Der Assembler-Kurs »Assembler ist keine Alchimie« hat die Grundlagen dieser Sprache zwar vermittelt, aber bis zum eigenen Maschinenprogramm ist es häufig noch ein holpriger Weg. Ein Führer, der Ihnen den Pfad aus den weiten Basic-Ebenen auf die Gipfel der Assembler-Programmierung zeigt, ist dieser Kurs.

Zunächst halten wir uns an Bekanntes: Basic-Befehle sollen auf ihre Entsprechungen in Assembler untersucht werden. Oft aber streifen wir die Fesseln der Hochsprache ab und lernen allerlei Assembler-Techniken kennen. Wo es uns angemessen erscheint, nehmen wir uns die Freiheit, die computerinterne Firmware zu benutzen. Alles in allem sollen Sie durch Training zum vertrautem Umgang mit Assembler gelangen.

1. Basic contra Assembler

Basic macht uns den Umgang mit unserem Computer relativ leicht: Wir brauchen uns kaum um die Verwaltung von Variablen zu kümmern. Benötigen wir zum Beispiel eine neue Stringvariable, dann genügt ihr erstes Auftauchen und Basic richtet sie für uns gebrauchsfertig ein. Sehr entgegenkommend ist Basic auch, was den Verkehr mit Peripherie jeder Art angeht: Ein Zeichen von der Tastatur anzunehmen, ist mit GET oder INPUT recht einfach. Irgend etwas auf dem Drucker zu schreiben oder auf die Diskette: mit OPEN und dem PRINT # geht's reibungslos und ohne Gedanken an die Busverwaltung. Die Hochsprache nimmt uns vieles ab, worum wir uns in Assembler kümmern müssen.

Andererseits befindet man sich etwa in der Situation eines Bauherrn, der sich ein Haus mit Fertigteilen aufstellen lassen möchte: Jeder Wunsch, der vom Standard abweicht, ist nicht möglich oder wird sehr teuer. Basic-Erweiterungen sind dann mit einer großen Sammlung aller möglichen Fertigteil-Formen zu vergleichen: Man muß sie alle kaufen, obwohl man nur einige für den Sonderwunsch braucht.

Der Vergleich fängt nun etwas zu hinken an: Die Fertigteile setzen sich nämlich bei genauerem Hinsehen aus kleinen Einzelteilen (Ziegelsteinen) zusammen. Wenn man also auf die vorgefertigten Großteile weitgehend verzichtet, dafür aber den Umgang mit Ziegelsteinen (Assemblerbefehlen) beherrscht, kann man sich genau das individuelle Haus bauen, wie bizarr es auch immer aussehen mag.

Wenn man sich das Verarbeiten eines Basic-Programmes etwas genauer ansieht, dann versteht man die Schwerfälligkeit, mit der vieles geschieht. Vom Augenblick des Einschaltens an läuft ein Maschinenprogramm im Computer: Der Interpreter. Jeder Tastendruck, jeder Basic-Befehl, der diesem in die Hände fällt, wird untersucht und führt zur Abarbeitung eines auf diesen Befehl zugeschnittenen Assemblerprogrammes.

Ein Basic-Befehl wie beispielsweise PRINT kann ganz verschiedene Reaktionen erfordern: Wenn vorher ein CMD-Kommando erfolgt war, findet die Ausgabe nicht auf dem Bildschirm statt, wenn das nicht der Fall war, kann der Bildschirm an einer anderen Stelle als im Normalfall liegen. Das, was auszudrucken ist, kann ein String sein, eine Integerzahl oder eine Fließkommavariable, allerlei Fehlerquellen sind abzufangen etc. Das zu PRINT gehörende Assembler-Programm muß all diese Möglichkeiten berücksichtigen, wie selten sie auch angesprochen werden.

Bei einem individuellen Assembler-Programm wissen wir dagegen, was wir wie ausgeben wollen. Unser eigener »PRINT«-Befehl wird nur das enthalten, was unbedingt notwendig ist, der ganze unnötige Ballast wird von uns nicht programmiert. Allerdings sind wir dann auch voll verantwortlich für die einwandfreie Funktion. Wir müssen beispielsweise dafür sorgen, daß bei Zugriffen auf Register oder Speicher dort der richtige Wert zum richtigen Zeitpunkt im erlaubten Format zur Verfügung steht. Das ist wie im täglichen Leben: Die größere Freiheit legt uns mehr Verantwortung auf und schafft uns andererseits ungeahnte Möglichkeiten, Ideen zu verwirklichen.

2. Noch eine technische Vorbemerkung

Für alle Beispielprogramme werde ich den Assembler Hypra-Ass von Gerd Möllmann verwenden. Er wurde im 64'er, Ausgabe 7/85, Seite 66, als Listing des Monats abgedruckt und ist auch auf der Leserservice-Diskette erhältlich. In dieser Preislage (6.50 Mark und etwas Schweiß beim Eintippen, beziehungsweise 29,80 Mark für die Diskette) habe ich noch keinen besseren gesehen und auch fast alle kostspieligeren Assembler reichen ihm nicht das Wasser. Weil man aber auch einen Monitor braucht, findet zu diesem Zweck weiterhin der SMON hier seinen Platz. Ich verwende eine Version, die bei $9000 beginnt, um ab $C000 Raum zu lassen für den Reassembler zu Hypra-Ass (64'er, Ausgabe 11/85). Das ist ein nützliches Programm, mit dem man Maschinencode aus dem Speicher wieder in einen Quelltext umwandeln kann. Mittels des Hypra-Ass ist dieser Quelltext dann bearbeitbar. Mit diesem kompletten Instrumentarium sind wir allen Aufgaben gewachsen.

3. Einfache Schleifen

Eine der meistgebrauchten Strukturen in Basic und auch eine der wichtigsten in Assembler ist die Programmschleife. Als »einfache« Schleife bezeichne ich solche, die zum Zählen nur 1 Byte erfordern, also maximal 256 Durchläufe erlauben. Die verschiedenen Möglichkeiten sehen wir uns anhand von Verzögerungsschleifen an, die zunächst einmal nichts anderes tun, als zu zählen und Zeit zu verbrauchen (welch ein Luxus!). Die einfachste Variante lautet in Basic etwa:

```
10 FOR I = 0 TO 255
20 NEXT I
```

In Bild 1 finden Sie die »Übersetzung« in Assembler.

Sie können sowohl das Y-Register (Variante 1) als auch das X-Register (Variante 2) zum Zählen verwenden. In Zeile 5 finden Sie .LI 1,4. Das ist ein Pseudobefehl — also kein 6502-Befehl —, der die Ausgabe des Protokolls über den Drucker bewirkt. Zeile 40 enthält durch .BA $5000 wieder einen Pseudobefehl. Damit legt man fest, von welcher Adresse an der Maschinencode in den Speicher gelegt werden soll. Die Zeilen 50 bis 90 sind unser Assemblerprogramm. Zuerst wird ein Startwert 0 in das Y-Register geschrieben und dieses dann in der Zeile mit dem Label um 1 hochgezählt.

Falls Ihnen der Ausdruck Label noch nicht geläufig ist: Natürlich kann man auch statt dessen die Adresse 5002 hinter den BCC-Befehl in Zeile 80 schreiben — so haben wir das ja bisher immer mit dem SMON-Assembler getan. Das hätte aber im Quelltext, den wir hier schreiben, den Nachteil, daß wir diese Adresse jedesmal ändern müßten, wenn wir uns entschlössen, mit dem Pseudobefehl .BA den Programmstart zu verlegen. Indem wir aber diese Zeile durch das Label-Kennzeichen markieren, merkt sich der Hypra-Ass die dazugehörige Zeilennummer und rechnet sie beim Assemblieren automatisch in die richtige Sprungadresse um.

Ein weiterer Vorteil ist, daß man zu Dokumentationszwekken jede wichtige Adresse auf diese Weise markieren und sich am Schluß durch eine Symboltabelle ausgeben lassen kann. Besonders bei langen Programmen, in denen man dann sinnvolle Labelnamen verwendet (beispielsweise DRUCKEN am Anfang des Programmteils, das einen Ausdruck steuert), kann das eine unschätzbare Hilfe sein.

In unserem Programm in Bild 1 geht es weiter mit dem Vergleich, ob im Y-Register nach der Erhöhung schon $FF erreicht wurde. Ist das nicht der Fall, dann ist das Carry-Bit frei und der Programmablauf verzweigt zurück zur Labelzeile. Ansonsten ist die Verzögerungsschleife beendet und mit dem BRK meldet sich der SMON, den Sie zu diesem Zeitpunkt natürlich im Speicher haben sollten (vergessen Sie nicht, den SMON zumindest einmal zu starten mit SYS »startadresse«, damit bei einem BRK in den SMON gesprungen wird).

Falls Sie das Programm durch SYS $5000 vom Hypra-Ass aus gestartet haben (und nicht durch G 5000 aus dem SMON), finden Sie sich ebenfalls im Monitor wieder. Fast alle Beispielprogramme in diesem Kurs werden mit BRK enden. Der Grund dafür ist, daß es oft interessant ist, die Register nach dem Programmende zu beobachten. Sollten Sie ohne Monitor arbeiten wollen, dann müßten Sie statt dessen ein

RTS einsetzen. Hinter dem eigentlichen Programm finden Sie .SY 1,4. Auch das ist ein Pseudobefehl, der die Ausgabe der Symboltabelle über den Drukker bewirkt. Nicht sichtbar ist ein Befehl .ST, mit dem die Assemblierung beendet wird. Einige interessante Angaben besorgt uns der Hypra-Ass noch nach der kurzen Symboltabelle: Eine Zeitangabe und den Bereich, in dem der Maschinencode nun nach der Assemblierung zu finden ist. Falls Sie diesen Objektcode (so nennt man den Maschinencode auch häufig) speichern wollen (vom Monitor aus mit dem S-Kommando möglich), dann brauchen Sie diese Angaben. Unser Programm würde dann so abgespeichert:
S"OBJ.VERZ.VAR1" 5000 5008
(Man muß immer ein Byte zur Endadresse hinzurechnen beim Speichern des Objektcode). Eine andere Möglichkeit, den Objektcode auf Diskette zu speichern: Nach dem .LI 1,4 in der nächsten Zeile folgenden Befehl einsetzen:
10 -.OB"OBJ.VERZ.VAR1,p,w
Jetzt wird nach dem Starten des Assemblierens mit RUN automatisch das Maschinenprogramm gespeichert.

Die weiteren Programmbeispiele werde ich nicht so erschöpfend erklären wie dieses. Nur wenn neue Pseudobefehle verwendet werden oder eine neue Programmstruktur das erfordert, geht's nochmal in die Tiefe. Um etwas Platz zu sparen, wurden die folgenden Programme nicht mit dem .LI - Befehl aus dem Drucker ausgegeben, sondern mit
OPEN 1,4:CMD1
/E
Dadurch werden die Adressen mit den Hex-Codes der Maschinenbefehle nicht gedruckt, sondern nur das Listing, wie es auch auf dem Bildschirm zu sehen ist.

Häufig tritt in Schleifen der Fall ein, daß weder das Y- noch das X-Register zur Verfügung stehen. Sie dienen dann anderweitig schon als Index. Statt dessen kann ebensogut eine Speicherstelle den Zähler bilden, wie in dieser Variante 3:

```
      LDA  #$00
      STA  $FB    ;$FB ist Zähler
LABEL INC  $FB
      LDA  $FB
      CMP  #$FF
      BCC  LABEL
      RTS
```

Selbstverständlich kann auch jede andere Speicherstelle anstelle von $FB verwendet werden, sogar eine, die nicht in der Zeropage liegt. Voraussetzung ist lediglich, daß sie nicht innerhalb der Schleife verändert wird — außer zum Zählen der Schleifendurchläufe. In den bisher kennengelernten Varianten haben wir immer $FF als Endwert genommen. Nun steht man oft vor der Aufgabe, bis zu einem bestimmten Endwert zu zählen, der vorher irgendwie eingegeben oder festgelegt wird. In Basic sähe das beispielsweise so aus:

```
HYPRA-ASS  ASSEMBLERLISTING:

              5    -          .LI 1,4
;*** VERZOEGERUNGSSCHLEIFE VARIANTEN 1 UND 2 ***
; X- ODER Y-REGISTER ALS ZAEHLER
;
              40   -          .BA $5000
5000 A000    :50   -          LDY #$00     ;BZW. LDX
5002 C8      :60   -LABEL     INY          ;     INX
5003 C0FF    :70   -          CPY #$FF     ;     CPX
5005 90FB    :80   -          BCC LABEL    ;WENN <255
5007 00      :90   -          BRK
              100  -          .SY 1,4

SYMBOLS IN ALPHABETICAL ORDER:

LABEL         = $5002

END OF ASSEMBLY  0:14.6
BASE = $5000  LAST BYTE AT $5007
```

Bild 1. Etwas zögern mit Variante 1 und 2

```
10   -           .LI 1,4
20   -           .BA $5000
30   -;*** VERZOEGERUNGSSCHLEIFE VARIANTE 4 ***
40   -;VARIABLER ENDWERT
50   -;Y-REGISTER ALS ZAEHLER
60   -;
70   -           LDA #$20       ;DAS IST DEZIMAL 32
80   -           STA $FA        ;ENDWERT SPEICHERN
90   -;
100  -           LDY #$00       ;Y-REGISTER INITIALISIEREN
110  -LABEL      INY
120  -           CPY $FA        ;ENDWERT ERREICHT?
130  -           BCC LABEL      ;NEIN:DANN WEITERZAEHLEN
140  -           BRK
150  -;
160  -           .SY 1,4
170  -           .ST
```

Bild 2. Verschieden zögern mit Variante 4

```
10   -           .LI 1,4
20   -           .BA $5000
30   -;*** VERZOEGERUNGSSCHLEIFE VARIANTE 5 ***
40   -;VARIABLER ENDWERT
50   -;SPEICHERSTELLE ALS ZAEHLER
60   -;
70   -           LDA #$20       ;DAS IST DEZIMAL 32
80   -           STA $FA        ;ENDWERT SPEICHERN
90   -;
100  -           LDA #$00       ;ZAEHLER INITIALISIEREN
110  -           STA $5100      ;DA IST ER:UNSER ZAEHLER
120  -;
130  -LABEL      INC $5100
140  -           LDA $5100
150  -           CMP $FA        ;VERGLEICHEN MIT ENDWERT
160  -           BCC LABEL      ;WEITERZAEHLEN WENN <> ENDWERT
170  -           BRK
180  -           .SY 1,4
190  -           .ST
```

Bild 3. Das ist die Variante 5

```
10   -           .LI 1,4
20   -           .BA $5000
30   -;*** VERZOEGERUNGSSCHLEIFE VARIANTE 6 ***
40   -;ABWAERTS ZAEHLEN (Y-REGISTER)
50   -;
60   -           LDY #$FF       ;STARTWERT NACH Y
70   -LABEL      DEY
80   -           BNE LABEL      ;WEITER BIS Y = 0
90   -           BRK
100  -;
110  -           .SY 1,4
120  -           .ST
```

Bild 4. Rückwärts zögern mit Variante 6

```
10   -           .LI 1,4
20   -           .BA $5000
30   -;*** VERZOEGERUNGSSCHLEIFE VERSION 7 ***
40   -;ABWAERTS ZAEHLEN ($FA ALS ZAEHLER)
50   -;
60   -           LDA #$20       ;STARTWERT IN ZAEHLER
70   -           STA $FA        ;SCHREIBEN
80   -LABEL      DEC $FA
90   -           BNE LABEL      ;WEITER BIS $FA = 0
100  -           BRK
110  -;
120  -           .SY 1,4
130  -           .ST
```

Bild 5. Da haben wir die Variante 7

```
10 A = 32
20 FOR I = 0 TO A
30 NEXT I
```

Hier ist also der Endwert in Zeile 10 auf 32 gesetzt worden und die Schleife zählt bis zu diesem in A fixierten Wert. In Assembler können wir das ebenfalls. Bild 2 zeigt die Variante 4.

Die Speicherstelle $FA nimmt die Funktion der Variablen A des Basicprogrammes ein. Dorthinein wird der Endwert (32 = $20) gelegt und der Vergleichsbefehl lautet nun:
CPY $FA
Das ist: »Vergleiche den Inhalt des Y-Registers mit dem Inhalt der Speicherstelle $FA«. Wir haben in dieser Version 4 wieder das Y-Register als Zähler benutzt, Version 5 zeigt uns in Bild 3 dasselbe, nur wird hier die Speicherstelle $5100 zum zählen verwendet.

Es hat sich eingebürgert, Schleifen in Assembler nicht — wie wir es bisher getan haben — aufwärts, sondern sie abwärts zu zählen. Der Grund dafür ist: Es geht schneller, weil man sich meistens den Compare-Befehl ersparen kann. Bei Verzögerungsschleifen ist das ja noch nicht so interessant, später aber, wenn in den Schleifen noch allerhand geschehen soll, summieren sich die Taktzeiten bei mehrfachen Durchlauf schon ganz erheblich. Eine Basic-Programmsequenz sähe nun so aus:

```
10 FOR I = 255 TO 0 STEP -1
20 NEXT I
```

In Bild 4 finden Sie das Assemblerlisting der Variante 6.

Das entspricht der Variante 1. Der Unterschied ist aber, daß hier abwärts gezählt wird und man sich den CPY-Befehl sparen kann, denn vor einem Unterlauf des Y-Registers wird automatisch bei 0 die Zero-Flagge gesetzt. Das aber prüft der BNE-Befehl.

Aus alledem ist also zu lernen:
1) Wann immer möglich, abwärts zählen.
2) Wann immer möglich, X- oder Y-Register als Zähler verwenden.

Die Variante 5 war natürlich ein ausgesuchtes Extrembeispiel, denn außer der Tatsache, daß man beim Abwärtszählen den Endwert als Startwert immer gleich in den Zähler eingeben kann und ihn normalerweise nicht noch irgendwo speichern muß, verwendet man natürlich — wenn es denn nötig ist, etwas anderes als die Indexregister dazu zu gebrauchen — eine Zeropagespeicherstelle als Zähler und nicht — wie in Version 5 — eine Speicherstelle wie $5100. Die verbesserte Version 7 entspricht dem Basicprogramm:

```
10 A = 32
20 FOR I = A TO 0 STEP -1
30 NEXTI
```



In Bild 5 finden Sie diese Version. $FA dient als Zähler.

Etwas schwieriger wird die Programmierung, wenn man nicht nur um 1 herauf- oder herunterzählt, sondern um 2,3,4 oder mehr. Das Basic-Äquivalent drückt sich dann beispielsweise in der Ergänzung STEP -2 der FOR...NEXT-Schleife aus. Dreht es sich nur um kleine Schrittweiten, die konstant bleiben, dann verwendet man vorteilhaft mehrere DEY (oder DEX, DEC, INY, INX und INC) hintereinander. Man muß außerdem mit der Abbruchbedingung einer solchen Schleife vorsichtig sein. BNE ist nicht immer möglich, weil man unter Umständen schon vor der Prüfung (durch BNE) unter 0 hindurch gezählt hat (dann folgt ja wieder $FF etc.). Hat beispielsweise der Zähler (hier das Y-Register) den Wert 1 und es werde durch eine Sequenz:

```
DEY
DEY
DEY
BNE LABEL
```

weitergezählt, dann nimmt Y der Reihe nach die Werte 0,FF,FE an und BNE findet die Zeroflagge nicht gesetzt. Man muß also andere Abbruchbedingungen verwenden. Solange man bis zur ersten Prüfung (also dem ersten Schleifendurchlauf beim Herunterzählen) im Zähler mindestens $7F (=binär 0111 1111) vorliegen hat, kann man mittels BPL die Schleife schließen. Zur Erinnerung: BPL verzweigt, wenn Bit 7 nicht gesetzt ist (kleiner 128), BMI verzweigt, wenn Bit 7 gesetzt ist (größer oder gleich 128). Das Basic-Programmstück

```
10 FOR I = 32 TO 0 STEP -3
20 NEXT I
```

findet seine Entsprechung in dem Assemblerlisting Version 8 in Bild 6.

Wieder dient das Y-Register als Schleifenzähler.

Größere Schrittweiten lassen es — von einer gewissen Grenze an, die durch das Verhältnis von Bytezahl auf der einen und Bearbeitungsdauer auf der anderen Seite, bestimmt wird — sinnvoll erscheinen, den Zähler durch Subtraktion (oder Addition beim Aufwärtszählen) zu verändern. Das Analogon zur Basic-Sequenz:

```
10 FOR I = 127 TO 0 STEP -10
20 NEXT I
```

sehen Sie in Bild 7.

In dieser Version 9 dient die Zeropagespeicherstelle $FA als Zähler und in den Programmzeilen 80 bis 110 findet die Verminderung dieses Zählers durch Subtraktion statt ($0A = dezimal 10). Das Programm kann noch verändert werden, indem man anstelle von BPL den BCS-Befehl verwendet. Wenn die Subtraktion einen Unterlauf ergeben hat, wird das Carry-Bit gelöscht.

```
10   -             .LI 1,4
20   -             .BA $5000
30   -;*** VERZOEGERUNGSSCHLEIFE VERSION 8 ***
40   -;FOR I = 32 TO 0 STEP-3
50   -;
60   -             LDY #$20        ;STARTWERT IN ZAEHLER
70   -LABEL        DEY             ;MINUS 3
80   -             DEY
90   -             DEY
100  -             BPL LABEL       ;WEITER BIS UNTERLAUF
110  -             BRK
120  -;
130  -             .SY 1,4
140  -             .ST
```

Bild 6. Verzögern in kleinen Schritten mittels Variante 8

```
10   -             .LI 1,4
20   -             .BA $5000
30   -;*** VERZOEGERUNGSSCHLEIFE VERSION 9 ***
40   -;FOR I = 127 TO 0 STEP -10
50   -;
60   -             LDA #$7F        ;DAS IST DEZIMAL 127
70   -             STA $FA         ; UNSER ZAEHLER
80   -LABEL        SEC
90   -             LDA $FA
100  -             SBC #$0A
110  -             STA $FA
120  -             BPL LABEL       ;WEITER BIS UNTERLAUF
130  -             BRK
140  -             .SY 1,4
150  -             .ST
```

Bild 7. Verzögern in großen Schritten mit Variante 9

Außerdem lassen sich noch 2 Byte einsparen, indem man das STA $FA aus Zeile 110 herausnimmt und dafür das LABEL eine Zeile höher setzt. Allerdings geht das dann auf Kosten der Durchschaubarkeit unseres Programmes.

Wir wollen nun mit den einfachen Verzögerungsschleifen aufhören. Es gäbe noch weitere Aufgaben zu lösen (nämlich beispielsweise von einem bestimmten Startwert bis zu einem bestimmten Zielwert zu zählen), die vertraue ich aber Ihnen selbst an: Alles notwendige dazu können Sie aus unseren verschiedenen Versionen entnehmen und kombinieren. Interessant werden Schleifen hauptsächlich durch einen Job, der in ihnen wiederholt ausgeführt wird. Zwei Beispiele sollen uns zur Illustration in dieser Folge dienen. Vorher aber sollen noch einige Bemerkungen zur grundsätzlichen Architektur von Schleifen gemacht werden.

Im Prinzip setzt sich jede Schleife aus vier Bestandteilen zusammen:

Initialisierung. Beispielsweise wird hier der Startwert des Zählers festgelegt.

Verarbeitung. Das ist das, was in den Verzögerungsschleifen bisher leer blieb: Der Job.

Steuerung. Hoch- oder Herunterzählen des Zählers und Prüfen der Abbruchbedingung.

Ausgang. Das war bisher bei uns immer der BRK-Befehl.

Aus diesen vier Bestandteilen lassen sich zwei grundsätzliche Schleifenmöglichkeiten konstruieren, die Sie in Bild 8 dargestellt finden.

In Bild 8a haben wir das Prinzip vorliegen, das unseren normalen FOR...NEXT-Schleifen in Basic zugrundeliegt. Diese Schleife wird mindestens einmal durchlaufen. Erst nach der Ausführung des Jobs erfolgt die Prüfung, ob die Abbruchbedingung gegeben war. Soll solch eine Schleife n-mal durchlaufen werden, muß die Initialisierung mit n-1 im Zähler erfolgen (oder die Abbruchbedingung entsprechend umgeformt werden).

Die Schleifenkonstruktion in Bild 8b dagegen muß nicht durchlaufen werden. Ihr entspricht etwa eine DO UNTIL...LOOP-Schleife oder auch eine DO WHILE...LOOP-Schleife aus dem Basic 7.0 des C 128. Hier erfolgt die Initialisierung des Zählers genau mit dem Wert n.

Sehen wir uns beispielsweise unsere Version 9 an, dann entdecken wir die einzelnen Schleifenteile wie folgt:

Initialisierung:	**LDA #$7F**
	STA $FA
Verarbeitung:LABEL	—
Steuerung:	**SEC**
	LDA $FA
	SBC #$0A
	STA $FA
	BCS LABEL
Ausgang:	**BRK**

Auf diese Weise ist es Ihnen möglich, alle bisher kennengelernten Schleifenvarianten mit einem beliebigen Job zu füllen. Noch eines gibt es zu bedenken: Alle Instruktionen zwischen dem Label und der Abbruchbedingung werden oft ausgeführt, sind also zeitfressend. Daher sollte der auszuführende Job alle Befehle vermeiden, die ebensogut vor der eigentlichen Schleife stehen könnten.

Sehen wir uns unser 1. Beispiel an. Wir stellen uns die Aufgabe, von den 127 Zeichen, die mittels POKE-Code erfaßbar sind, jedes 2. Zeichen an jeder 2. Bildschirmstelle abzubilden. Das Ganze soll durch Verwenden verschiedener Farben auch noch hübsch bunt aussehen. In Basic würden wir dafür schreiben:

```
10 S = 1024 : C = 55296
20 FOR I = 127 TO 0 STEP -2
30 POKE S+I,I
40 POKE C+I,I
50 NEXT I
```

Weil im Bildschirmfarbspeicher nur die Bits 0 bis 3 eine Rolle spielen (die anderen aber gar nicht beachtet werden), erzeugen wir in Zeile 40 auch die verschiedenen Farben mehrmals nacheinander.

(Heimo Ponnath/gk)

Fortsetzung folgt.

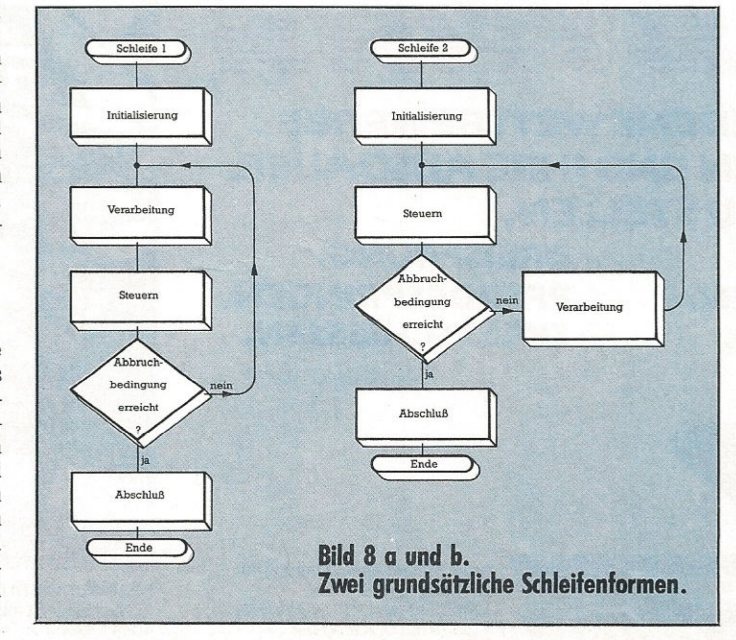


Bild 8 a und b. Zwei grundsätzliche Schleifenformen.



Aufgebohrt

Hier wird beschrieben wie man die 1541 zu einem IEEE-Bus-Laufwerk umrüstet und dreifache Geschwindigkeit bei allen Floppy-Befehlen erreicht.

Die 1541-Floppy ist ein abgemagertes 2031/4031-Laufwerk. Was steht also näher, sie wieder in den Originalzustand zu versetzen. Mit dieser Erweiterung sind zwar keine Übertragungsgeschwindigkeiten wie etwa bei ProLogic-DOS zu erreichen, aber immerhin werden alle Zugriffe auf die Diskette dreimal schneller. Die Geschwindigkeitssteigerung wird durch den parallelen IEC-Bus erreicht. Das 4040-Doppellaufwerk ist bekanntlich bis zu 6mal schneller, hier sind dafür aber zwei Prozessoren beschäftigt, der eine zum Lesen, der andere zum Übertragen der Daten. Mit dem einen Prozessor der 1541 lassen sich trotzdem ganz vernünftige Werte erzielen. Für den Anschluß am C 64 benötigen Sie ein IEC-Bus-Interface, wie wir es in Ausgabe 7/85, Seite 44 beschrieben ist. Ihre 1541 versteht nach Umbau die »Sprache« der großen CBM-Geräte. Sie können sie also auch direkt an einen CBM 3032/8032-Rechner anschließen und erhalten dadurch eine preisgünstige Alternative zum Doppellaufwerk.

Die Hardware

Um den parallelen Bus zu realisieren wird einer der beiden Portbausteine VIA 6522 benötigt. Dieser VIA-Baustein (Versatile Interface Adapter) ist teilweise nicht beschaltet. Um der IEC-Norm entsprechen zu können, sind acht Datenleitungen, sowie fünf Handshake-Leitungen zum Verbindungsaufbau erforderlich. Über die Handshake-Leitung teilt der Computer dem Laufwerk mit, daß er Daten senden, beziehungsweise empfangen möchte (ATN), daß die Daten gültig sind (DAV), daß keine Daten empfangen werden können (NRFD), daß die Daten noch nicht verarbeitet wurden (NDAC), oder daß die Übertragung beendet wurde (EOI). Wenn der Computer eine solche Übertragungssequenz beginnt, wird zuerst festgestellt, ob das Peripheriegerät überhaupt vorhanden beziehungsweise eingeschaltet ist (DEVICE NOT PRESENT). Hierfür sind die Leitungen »NRFD« und »NDAC« zuständig. Wird »ATN« gesetzt, so müssen beide Leitungen sofort nach Masse gezogen werden. Diese Funktion übernimmt das TTL-Gatter (EX-OR 74LS136) in der Schaltung. Die IEC-Norm schreibt weiterhin »Open-Collector«-Ausgänge vor, um mehrere Geräte anschließen zu können, das heißt die Leitungen sind im aktiven Zustand auf Masse-Potential (Low). Die 6522-VIA (Portbelegung siehe Bild 1) hat aber TTL-Pegel. Die notwendige Anpassung übernehmen die beiden IEC-Treiber 75160/161.

Wenn die Platine mit allen Bauteilen bestückt ist, wird die Leiterplatte anstelle der VIA in die Floppy gesteckt und die VIA von oben aufgesetzt. Das serielle Kabel hat in diesem Fall keine Funktion mehr, da alle Übertragungen nur noch parallel stattfinden.

Die Software

Das Floppy-DOS ist in zwei ROMs untergebracht, das $C000-ROM und das $E000-ROM. Um die Hardware bedienen zu können, müssen die seriellen Routinen durch parallele ersetzt werden. Bei den Laufwerken 2031 und 4031 ist davon der gesamte Bereich betroffen; alle Adressen sind um zirka 40 Byte verschoben. Das hat zur Folge, daß bei der 2031/4031 einige Programme mit Direkteinsprüngen im DOS nicht das vorfinden, was sie erwarten. Außerdem müßten beide ROMs gewechselt werden.

Bei unserem IEC-DOS bleibt der $C000-Bereich unangetastet. Die entsprechenden Routinen sind am Originalplatz eingebunden ($E853), sind aber etwas umfangreicher und belegen zusätzlich noch einige Bytes bei $E780 und $FF10. Bei dieser Gelegenheit wurde die Kopfbewegung verbessert und einige kleine Fehler im DOS beseitigt. Die Formatierungsroutine wurde im Originalzustand belassen.

Das Listing

Das Generatorprogramm im Listing erwartet ein 1541-Laufwerk. Nach dem Starten wird das Original-DOS ausgelesen und ab $5000 im C 64 abgelegt. Die Veränderungen werden anschließend mittels POKE-Schleife vorgenommen. Zum Schluß wird die aktuelle Prüfsumme berechnet und eingetragen.

Das Programm fragt, ob das veränderte DOS abgespeichert werden soll. Wird diese Frage mit »J« beantwortet, so erzeugt es ein File mit dem Namen: 1541.IEEE-DOS.

Sie können jetzt den Bereich von $5000 bis $6FFF in ein 2764-EPROM brennen und es mittels Kernel-Adapter (64'er, Ausgabe 4/85) anstelle des 1541-ROMs in den Steckplatz UB4 einsetzen. In manchen Laufwerken befindet sich in UB4 bereits ein EPROM mit dazugehörigem Adapter.

Der Zusammenbau

Den Schaltplan, den Bestückungsplan sowie das Platinenlayout im Maßstab 1:1 finden Sie in Bild 2 bis 4. Bitte beachten Sie, daß die Bestückungsseite die Platine von oben, und die Lötseite die Platine von unten zeigt. Aus Bild 5 entnehmen Sie bitte die Bauteileliste. Die Platine wurde einseitig ausgeführt. Es empfiehlt sich, alle Bausteine zu Sockeln, um sie bei eventuellen Beschädigungen leicht auswechseln zu können, in jedem Fall muß aber der 6422 gesockelt sein. Die Position der Bauteile entnehmen Sie dem Bestückungsplan. Zuerst werden die Widerstände R1 + R2 und die Drahtbrücke eingelötet. Bei den Dioden D1, D2 und D3 muß auf die Einbaurichtung geachtet werden. Der Pfeil zeigt auf die Anode, was zumeist durch einen Ring gekennzeichnet wird. Anschließend werden die Sockel verlötet. Um die Schaltung in den VIA-Sockel UC3 einsetzen zu können, müssen zwei 20polige Pinleisten von der Lötseite auf die Platine gelötet werden. Dieser Vorgang erfordert besondere Sorgfalt damit keine unerwünschten Lötbrücken entstehen. Zuletzt ist nur noch das IEEE-Anschlußkabel einzulöten und die Bausteine können eingesetzt werden. Um den Kabelanschluß zu vereinfachen kann ein Einpreßstecker mit Flachbandkabel verwendet werden. Die Anschlußbohrungen sind bereits nach IEEE-Norm angeordnet (siehe Bild 6).

Die beiden Brücken auf der Leiterbahnseite sind für die Geräteadresse zuständig. Ihre Funktion ist analog zu den Brücken des 1541-Laufwerks, um Adresse 8 einzustel-



len, sind beide geschlossen.

Bei älteren Laufwerken sind die Bausteine UC3 und UB4 nicht immer an derselben Stelle, auch die Bezeichnung variiert.

Es empfiehlt sich also vor dem Einbau einen Fachmann zu Rate zu ziehen, vor allem, wenn man sich bei seiner Laufwerksversion nicht ganz sicher ist. Auf keinem Fall sollte man das Prinzip von Versuch und Irrtum anwenden wollen, sonst kann eine Beschädigung der Laufwerkslogik nicht ausgeschlossen werden.

Der komplette Bausatz wird ab der nächsten Ausgabe über den Hardware-Service erhältlich sein, beachten Sie dazu die Hinweise auf den Service-Seiten.

(Michael Lamm/og/aw)

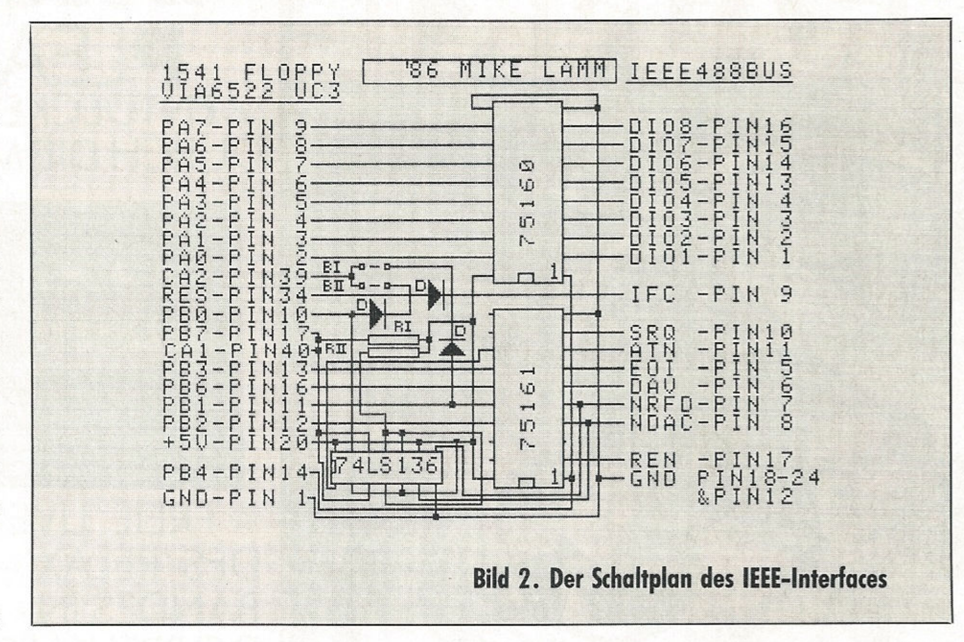


Bild 2. Der Schaltplan des IEEE-Interfaces

Platine	AMP-Stecker	TRW-Stecker	Signal
Pin 1	1	1	DIO 1
Pin 2	2	2	DIO 2
Pin 3	3	3	DIO 3
Pin 4	4	4	DIO 4
Pin 5	5	5	EOI
Pin 6	6	6	DAV
Pin 7	7	7	NRFD
Pin 8	8	8	NDAC
Pin 9	9	9	IFC
Pin 10	10	10	SRQ*
Pin 11	11	11	ATN
Pin 12	12	12	GND
Pin 13	13	A	DIO 5
Pin 14	14	B	DIO 6
Pin 15	15	C	DIO 7
Pin 16	16	D	DIO 8
Pin 17	17	E	REN*
Pin 18–24	18–24	F–N	GND

* Diese Signale werden bei Commodore nicht benutzt.

Bild 6. Die Beschaltung des IEEE-Kabels

DRA $1800/DDRA $1802
PA 0—PA 7 = DIO 0—DIO 7

DRB $1801	DDRB $1803
PB 0	ATN A
PB 1	NRFD
PB 2	NDAC
PB 3	EOI
PB 4	T/R
PB 5	frei
PB 6	DAV
PB 7	ATN
CA 1	ATN
CA 2	Device-Nr.

Bild 1. Portbelegung des 6522

1	* 2764 EPROM
1	* SN 75160 (Texas)
1	* SN 75161 (Texas)
1	* 74 LS 136
3	* 1 N 4148 Diode
2	* 3K3 ¼ Watt
1	Sockel 14pol.
2	Sockel 20pol.
1	Sockel 40pol.
2	Pinleisten 20pol.
1	Steckadapter f. EPROM
1	Meter Flachbandkabel
1	IEEE-488 Stecker AMP 24pol. oder
1	TRW User-Port-Stecker
1	Platine

Bild 5. Stückliste zum IEEE-Interface

```
10 POKE 45,0:POKE 46,113:CLR                       <000>
20 PRINT"{CLR}IEEE-ROM GENERATOR                   <068>
30 PRINT"{DOWN}C1986 BY MIKE LAMM                  <075>
40 PRINT"{DOWN}BITTE WARTEN......                  <058>
50 PRINT                                           <152>
60 F$="1541.IEEE-DOS                               <117>
70 OPEN 15,8,15,"I"                                <125>
80 INPUT#15,DS,DS$,T,S,M:IF DS THEN PRINT"
   DISK-ERROR:"DS;DS$,T,S:GOTO 320                 <059>
90 C=5*4096                                        <174>
100 FOR I=14*4096 TO 16*4096-1 STEP 128            <145>
110 PRINT#15,"M-R"CHR$(I-INT(I/256)*256);C
    HR$(INT(I/256));CHR$(128)                      <202>
120 FOR J=0 TO 127:GET#15,A$:POKE C,ASC(A$
    +CHR$(0)):C=C+1:NEXT J                         <033>
130 PRINT"*";                                      <173>
140 NEXT I:PRINT:PRINT                             <016>
150 READ A:READ B                                  <225>
160 FOR I=A TO A+B-1:READ D:POKE I,D:NEXT
    I                                              <114>
170 POKE 28678,44:POKE 28704,142:POKE 2870
    5,64:POKE 28706,3:SYS 7*4096                   <175>
180 IF PEEK(832)<>224 THEN PRINT"?UEBERTRA
    GUNGSFEHLER":GOTO 320                          <244>
190 PRINT"OVERLAY GESTARTET"                       <032>
200 RESTORE                                        <250>
210 FOR L=1 TO 10                                  <124>
220 READ A:READ B                                  <039>
230 FOR I=A TO A+B-1:READ D:POKE I,D:NEXT
    I,L                                            <026>
240 POKE 23746,8:POKE 25253,5:POKE 25257,2
    :POKE 25261,20                                 <047>
250 SYS 7*4096                                     <091>
260 PRINT:INPUT"ABSPEICHERN";A$                    <029>
270 IF A$<>"J"THEN 320                             <107>
280 POKE 193,0:POKE 194,80:POKE 174,0:POKE
     175,112                                       <163>
290 L=LEN(F$):FOR I=1 TO L:POKE I+831,ASC(
    MID$(F$,I,1)):NEXT                             <016>
300 POKE 183,L:POKE 187,64:POKE 188,3              <126>
```

Das Generatorprogramm; es erzeugt das IEEE-DOS. Bitte mit dem Checksummer eingeben.

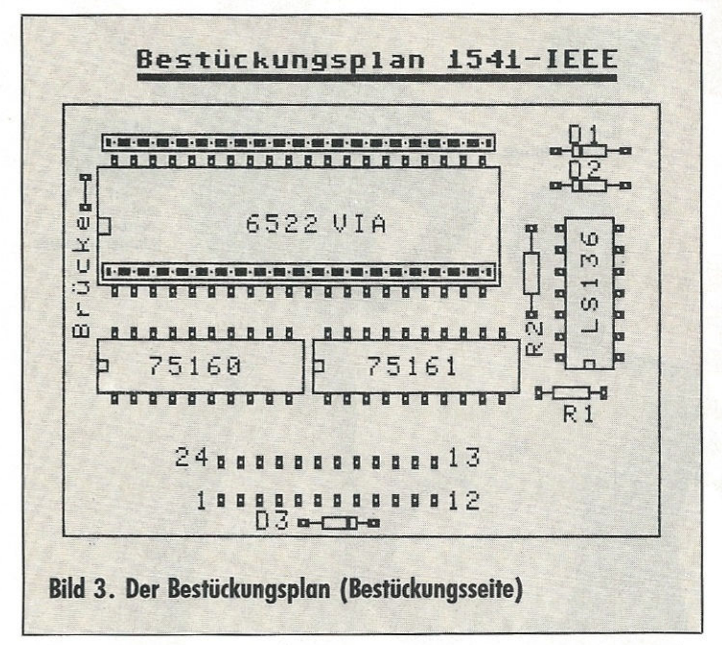


Bild 3. Der Bestückungsplan (Bestückungsseite)

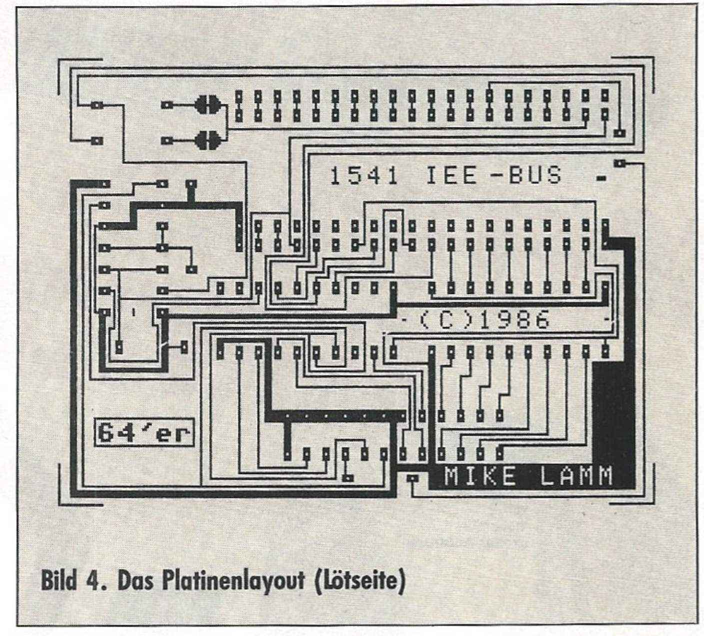


Bild 4. Das Platinenlayout (Lötseite)

```
310 SYS 62954                                    <020>
320 CLOSE 1:CLOSE 15:END                         <037>
330 REM PRUEFSUMMEN-ROUTINE                      <156>
340 DATA 28672,36                                <142>
350 DATA 162,112,134,35,169,31,141,230,110
    ,169,0,133,34,168,162,32,24,198,35           <228>
360 DATA 113,34,200,208,251,202,208,246,10
    5,0,170,73,255,141,230,110,96                <142>
370 REM $E5B7                                    <248>
380 DATA 21943,4                                 <145>
390 DATA 201,69,67,45                            <045>
400 REM $E5C4                                    <213>
410 DATA 21956,3                                 <111>
420 DATA 50,48,51                                <104>
430 REM $E780                                    <111>
440 DATA 22400,28                                <150>
450 DATA 96,142,1,24,142,3,24,169,206,141,
    0,24,169,49,141,2,24,169,104,141,0           <093>
460 DATA 28,169,111,141,2,28,96                  <113>
470 REM $E860                                    <055>
480 DATA 22624,526                               <031>
490 DATA 162,69,154,169,76,13,0,24,41,237,
    141,0,24,169,0,141,3,24,169,23,141           <222>
500 DATA 2,24,169,255,141,1,24,173,0,24,41
    ,251,9,3,141,0,24,44,0,24,80,4,48            <222>
510 DATA 249,16,125,169,253,45,0,24,141,0,
    24,41,8,133,248,173,1,24,73,255,133          <015>
520 DATA 150,169,4,13,0,24,141,0,24,160,0,
    165,150,41,96,201,64,240,41,201,32           <094>
530 DATA 240,6,201,96,240,47,208,70,165,15
    0,197,119,240,11,201,63,208,2,132            <042>
540 DATA 121,132,123,76,4,233,133,121,132,
    122,169,32,133,131,133,132,133,123           <027>
550 DATA 208,39,132,122,165,150,197,120,20
    8,229,133,122,132,121,240,232,165            <015>
560 DATA 123,240,21,165,150,133,132,72,41,
    15,133,131,104,41,240,201,224,208            <153>
570 DATA 5,88,32,192,218,120,44,0,24,80,25
    1,76,124,232,165,121,240,14,169,252          <239>
580 DATA 45,0,24,141,0,24,32,50,233,76,231
    ,235,169,250,45,0,24,9,4,141,0,24            <042>
590 DATA 165,122,240,3,32,220,233,76,231,2
    35,120,169,2,13,0,24,141,0,24,44,0           <013>
600 DATA 24,48,42,112,249,32,7,209,176,5,1
    81,242,106,176,83,165,132,41,240,201         <059>
610 DATA 240,240,75,165,131,201,1,240,19,4
    4,0,24,48,11,80,249,169,251,45,0,24          <189>
620 DATA 141,0,24,96,76,91,232,120,169,253
    ,45,0,24,141,0,24,169,4,13,0,24,141          <073>
630 DATA 0,24,44,0,24,48,231,80,249,173,0,
    24,41,251,141,0,24,169,2,13,0,24,141         <026>
640 DATA 0,24,44,0,24,48,208,80,249,76,109
    ,233,169,253,45,0,24,141,0,24,41,8           <086>
650 DATA 133,248,173,1,24,73,255,133,133,1
    20,169,4,13,0,24,141,0,24,44,0,24            <184>
660 DATA 48,170,80,249,169,251,45,0,24,141
    ,0,24,88,32,183,207,76,50,233,169            <195>
670 DATA 72,13,0,24,141,0,24,76,231,235,12
    0,32,235,208,176,6,166,130,181,242           <123>
680 DATA 48,1,96,169,89,141,2,24,169,255,1
    41[illegible],24,173,0,24,9,16,141,0,24,169  <013>
690 DATA 2,44,0,24,48,73,240,249,189,62,2,
    73,255,141,1,24,181,242,9,183,45,0           <247>
700 DATA 24,141,0,24,173,0,24,48,48,41,6,2
    01,6,240,176,41,2,208,241,88,32,170          <119>
710 DATA 211,120,169,4,44,0,24,48,26,240,2
    49,169,255,141,1,24,169,72,13,0,24           <136>
720 DATA 141,0,24,169,4,44,0,24,48,4,208,2
    49,240,151,76,91,232,173,0,24,9,64           <083>
730 DATA 141,0,24,96,169,2,13,0,24,41,254,
    141,0,24,96,170,170,170,170,170,170          <204>
740 DATA 170,170,170,170,170,170                 <010>
750 REM $EAA4                                    <055>
760 DATA 23204,3                                 <204>
770 DATA 32,129,231                              <193>
780 REM $EB3D                                    <251>
790 DATA 23357,14                                <154>
800 DATA 9,16,141,0,24,32,16,255,234,234,2
    34,234,234,234                               <094>
810 REM $EBDA                                    <220>
820 DATA 23514,22                                <112>
830 DATA 234,234,234,234,234,234,234,234,2
    34,234,32,128,231,88,234,234,234,234         <030>
840 DATA 234,234,234,234                         <030>
850 REM $F384                                    <169>
860 DATA 25476,8                                 <094>
870 DATA 3,133,74,165,34,32,72,255               <230>
880 REM $FF10                                    <101>
890 DATA 28432,72                                <111>
900 DATA 173,12,24,9,12,41,253,141,12,24,1
    73,2,24,41,254,141,2,24,173,0,24,41          <000>
910 DATA 3,9,72,133,120,73,96,133,119,173,
    12,24,9,2,141,12,24,173,0,24,41,238          <051>
920 DATA 141,0,24,173,2,24,9,1,141,2,24,96
    ,208,2,169,44,41,126,73,254,10,5,74          <126>
930 DATA 233,4,133,74,96                         <104>
```

© 64'er

Das Generatorprogramm für das IEEE-Interface (Schluß)

64'er SOFTWEAR-SERVICE

2
T-Shirt »64'er«
weiß,
4farbiger,
kleiner Aufdruck

3
T-Shirt »64'er«
hellblau,
4farbiger,
kleiner Aufdruck

5
Jogging-Anzug »64'er«
weiß,
1farbiger,
großer Aufdruck

4
Polohemd »64'er«
hellblau,
4farbiger,
kleiner Aufdruck

6
Jogging-Anzug »64'er«
graumeliert,
2farbiger,
großer Aufdruck

1
T-Shirt »64'er«
weiß,
4farbiger,
großer Aufdruck

Hallo, 64'er-Fans!

Nur für Euch gibt's jetzt T-Shirts, Polohemden und Jogging-Anzüge. Bestellen könnt Ihr gegen Voreinsendung des jeweiligen Betrags mit der Zahlkarte vom Programm-Service in diesem Heft. Bestellnummer bitte nicht vergessen – und falls vorhanden, Eure Kundennummer.

1. T-Shirt »64'er«
4farbiger, großer Aufdruck, 100% Baumwolle, weiter Schnitt, Jersey, Farbe: weiß

Größe 4 = S	Best.-Nr. TS 104S	DM 19,90
Größe 5 = M	Best.-Nr. TS 105M	DM 19,90
Größe 6 = L	Best.-Nr. TS 106L	DM 19,90
Größe 7 = XL	Best.-Nr. TS 107XL	DM 19,90

2. T-Shirt »64'er«
4farbiger, kleiner Aufdruck, 100% Baumwolle, weiter Schnitt, Jersey, Farbe: weiß

Größe 4 = S	Best.-Nr. TS 114S	DM 19,90
Größe 5 = M	Best.-Nr. TS 115M	DM 19,90
Größe 6 = L	Best.-Nr. TS 116L	DM 19,90
Größe 7 = XL	Best.-Nr. TS 117XL	DM 19,90

3. T-Shirt »64'er«
4farbiger, kleiner Aufdruck, 100% Baumwolle, weiter Schnitt, Jersey, Farbe: hellblau

Größe 4 = S	Best.-Nr. TS 124S	DM 22,50
Größe 5 = M	Best.-Nr. TS 125M	DM 22,50
Größe 6 = L	Best.-Nr. TS 126L	DM 22,50
Größe 7 = XL	Best.-Nr. TS 127XL	DM 22,50

4. Polohemd »64'er«
4farbiger, kleiner Aufdruck, 100% Baumwolle in Pique-Qualität, Farbe: hellblau, Lieferzeit ca. drei Wochen

Größe 4 = S	Best.-Nr. TS 134S	DM 49,90
Größe 5 = M	Best.-Nr. TS 135M	DM 49,90
Größe 6 = L	Best.-Nr. TS 136L	DM 49,90
Größe 7 = XL	Best.-Nr. TS 137XL	DM 49,90

5. Jogging-Anzug »64'er«
1farbiger, großer Aufdruck, 50% Baumwolle/50% Acryl, Farbe: weiß, Lieferzeit ca. drei Wochen

Größe 4 = S	Best.-Nr. TS 144S	DM 98,90
Größe 5 = M	Best.-Nr. TS 145M	DM 98,90
Größe 6 = L	Best.-Nr. TS 146L	DM 98,90
Größe 7 = XL	Best.-Nr. TS 147XL	DM 98,90

6. Jogging-Anzug »64'er«
2farbiger, großer Aufdruck, 50% Baumwolle/50% Acryl,
Farbe: graumeliert, Lieferzeit ca. drei Wochen

Größe 4 = S	Best.-Nr. TS 154S	DM 98,90
Größe 5 = M	Best.-Nr. TS 155M	DM 98,90
Größe 6 = L	Best.-Nr. TS 156L	DM 98,90
Größe 7 = XL	Best.-Nr. TS 157XL	DM 98,90

Größentabelle:	S	M	L	XL
Größe	4	5	6	7
Damen	38	40	42	44
Herren	46	48	50	52
Kinder	176			

Alle Artikel sind vom Umtausch ausgeschlossen!



Hans-Pinsel-Straße 2, 8013 Haar bei München

Der 64'er EPROM-Brenner

Weiter geht's mit unserem EPROM-Programmiergerät. Nach der vollständigen Bauanleitung können Sie nun seine Hardware nachbauen, uns so zu einem exklusiven EPROM-Brenner kommen.

Nachdem wir in der letzten Ausgabe die Funktionsweise des 64'er EPROM-Programmiergerätes rein theoretisch behandelt haben, werden wir in dieser Ausgabe eine ausführliche Bau- und Bedienungsanleitung für das Selbstbau-Programmiergerät bringen. Die zugehörige Software erscheint in der nächsten Ausgabe. Beginnen wir beim wichtigsten Teil des EPROM-Programmiergeräts, bei der Platine. Es handelt sich hierbei wegen der vielen Leiterbahnen (Adreß- und Datenbusse etc.) um eine doppelkaschierte (zweiseitige) Platine. Der Nachbau dieser Platine erfordert schon eine gewisse Portion an Erfahrung, da die beiden Platinenlayouts (Bild 1: Lötseite; Bild 2: Bestückungsseite) genau deckungsgleich sein müssen. Eine Ungenauigkeit von nur einem Millimeter kann die Platine schon unbrauchbar machen. Ein weiteres Problem werden die Durchkontaktierungen (Verbindungen zwischen beiden Platinenseiten) bereiten. Solche Durchkontaktierungen können im Regelfall nur von professionellen Platinenherstellern fehlerfrei angefertigt werden. Wer also zu wenig Erfahrung bei der Platinenherstellung hat, kann ab der nächsten Ausgabe die unbestückte Platine und den Spannungswandler beim 64'er Hardware-Service beziehen.

Beim Einlöten der Bauteile sind folgende Punkte zu berücksichtigen:

Zunächst ist darauf zu achten, einen geeigneten Lötkolben zu verwenden. Beim Umgang mit elektronischen Bauteilen, sollte man von der Verwendung des guten alten 300-Watt-Lötkolbens absehen, da es in der Regel wenig Sinn hat, die Bauteile zur Weißglut zu bringen. Optimal ist ein Lötkolben mit Trenntrafo und einer Leistung zwischen 15 und 30 Watt. Es ist außerdem geeignetes Lötzinn zu benutzen. In der Elektronik verwendet man hauptsächlich Röhrenlötzinn mit Flußmittel für säurefreie Lötstellen. Lötsäure, Lötfett oder Löthonig sind wegen deren Säureanteil in der Elektronik verboten!

Beim Bestücken der Platine sollte man eine bestimmte Reihenfolge einhalten. Zuerst werden die Widerstände eingelötet. Hier spielt die Richtung der Bauteile keine Rolle, da die Widerstände keine Polung besitzen. Anschließend werden die Kondensatoren eingesetzt und festgelötet. Hier ist zwischen normalen Kondensatoren und Elektrolytkondensatoren (Elkos) zu unterscheiden. Bei normalen Kondensatoren spielt die Polung keine Rolle. Elkos und Tantalelkos hingegen besitzen eine Elektrolytfüllung, um trotz kleiner Baugröße eine hohe Kapazität zu erreichen. Bei diesen Bauteilen ist unbedingt auf die Polung zu achten, da sie sich durch falsches Anschließen eventuell explosionsartig in Einzelteile auflösen. Um das zu vermeiden, sind Elkos eindeutig gekennzeichnet. Ein »+«-Zeichen oder eine kreisrunde Kerbe um das Kondensatorgehäuse (Sollbruchstelle bei Falschpolung) kennzeichnet den positiven Anschluß. Ein Minuszeichen oder ein Ring kennzeichnet den Minuspol. Im Schaltplan unterscheiden sich normale Kondensatoren von Elkos durch einen dickeren Balken mit »+«-Zeichen am Pluspol (Bild 3). Anschließend werden die Halbleiter eingelötet. Auch hier sollte

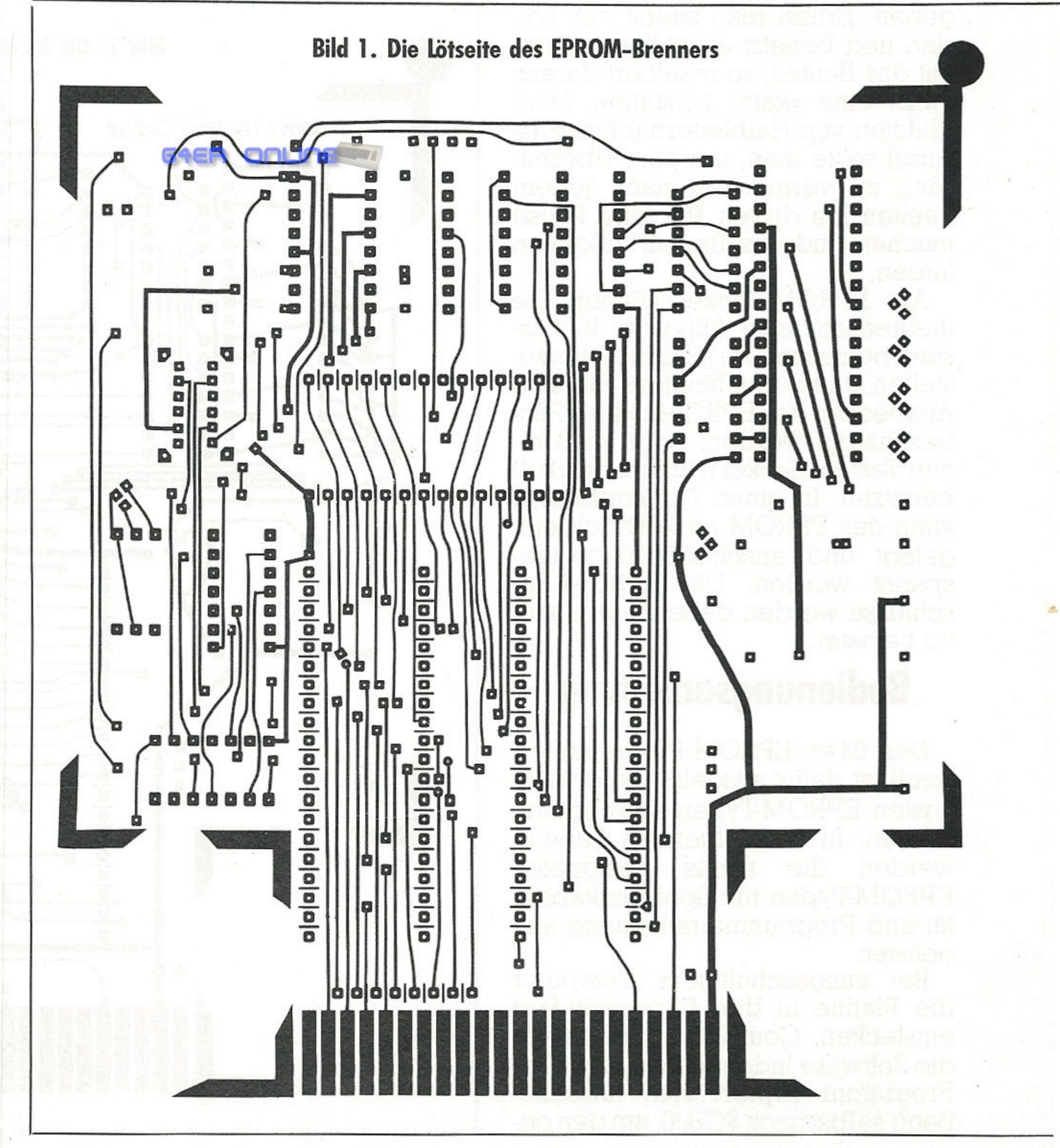


Bild 1. Die Lötseite des EPROM-Brenners

man eine bestimmte Reihenfolge einhalten. Zuerst die Dioden (auch Leuchtdioden), dann Transistoren und schließlich ICs.

Auch bei den Dioden ist auf die Polung zu achten. Die Kathode ist bei jeder Diode durch einen Ring am Diodengehäuse gekennzeichnet (Bild 4). Bei Leuchtdioden ist der Anschluß für die Anode kürzer als der Anschluß für die Kathode. Die Transistoranschlüsse können der Abbildung 4 entnommen werden. Das Platinenlayout ist jedoch so angefertigt, daß ein falsches Einlöten fast unmöglich ist. Sind alle anderen Bauteile (inklusive Schalter) verlötet, so werden die ICs eingelötet. Es empfiehlt sich vor allem bei den beiden I/O-Bausteinen (6821) Fassungen zu verwenden, um sie bei eventuellen Ausfällen leichter ersetzen zu können. Die Bauteileliste finden Sie in Bild 5, den Bestückungsplan in Bild 6. Auch beim Löten selbst sind einige wichtige Punkte zu beachten: Um möglichst saubere Lötstellen zu erhalten, sollte man zuerst das Bauteil erhitzen und anschließend Zinn zugeben. Erhitzt man zuerst das Lötzinn und benetzt anschließend damit das Bauteil, so resultiert daraus meist eine »kalte Lötstelle«. Beim Einlöten von Halbleitern (ohne Fassung) sollte man, um eine Überhitzung zu vermeiden, nach jedem zweiten bis dritten Pin eine Pause machen und das Bauteil abkühlen lassen.

Als EPROM-Sockel können Sie theoretisch jede 28polige IC-Fassung benutzen. Um jedoch sicherzustellen, daß durch häufiges Ein- und Ausstecken der EPROMs keine Pins beschädigt werden, sollte man einen Textool-Sockel (Nullkraftsockel) benutzen. In einen Nullkraftsockel kann das EPROM ohne Druck eingelegt und anschließend festgespannt werden. Die EPROM-Anschlüsse werden dabei nicht unnötig belastet.

Bedienungsanleitung

Das 64'er EPROM-Programmiergerät ist dafür ausgelegt, die wichtigsten EPROM-Typen zu programmieren. In der folgenden Tabelle werden die sechs geeigneten EPROM-Typen mit Speicherkapazität und Programmierspannung aufgelistet:

Bei ausgeschaltetem Computer die Platine in den Expansion-Port einstecken. Computer einschalten, die Software laden und starten. Das Programm kopiert sich anschließend selbst nach $C000, um den gesamten Basic-Speicher für die EPROM-Daten freizuhalten. Das 64-KByte-EPROM 27512 muß ohnehin in zwei Arbeitsgängen gebrannt werden, da dieses EPROM den ganzen adressierbaren Speicherbereich des C 64 aufnehmen könnte.

Typ	Speicher	Programmierspannung
2532	4 KByte	25 Volt
2732	4 KByte	25 Volt
2764	8 KByte	25 Volt
27128	16 KByte	12,5 Volt
27256	32 KByte	12,5 Volt
27512	64 KByte	12,5 Volt

Tabelle der möglichen EPROM-Typen

Nach kurzer Zeit erscheint am Bildschirm ein Menü mit folgenden Punkten:
(P)rogrammieren
(A)uslesen
(L)eertest
(V)erifizieren
(W)iederholen
(M)enü
(E)nde
(F)ile laden
(T)ype wählen

Der letzte Menüpunkt erscheint nur, wenn sich der Schalter auf dem Programmiergerät auf Stellung 27xxx befindet. Zeigt der Schalter vom Computer weg, so ist das EPROM-Programmiergerät auf den Typ 2532 eingestellt, wodurch sich der letzte Menüpunkt erübrigt. Befindet sich der Schalter in entgegengesetzter Position, so kann über den letzten Menüpunkt eines der restlichen EPROMs der 27er Reihe ausgewählt werden. Der gewählte EPROM-Typ wird rechts oben am Bildschirm angezeigt.

Anschließend sollten Sie das EPROM in den Textoolsockel einlegen. Dazu bringen Sie den Spannhebel am Sockel in senkrechte Position, legen das EPROM ein und spannen das EPROM durch Zurückklappen des Hebels fest. Beim Einlegen der EPROMs ist folgendes un-

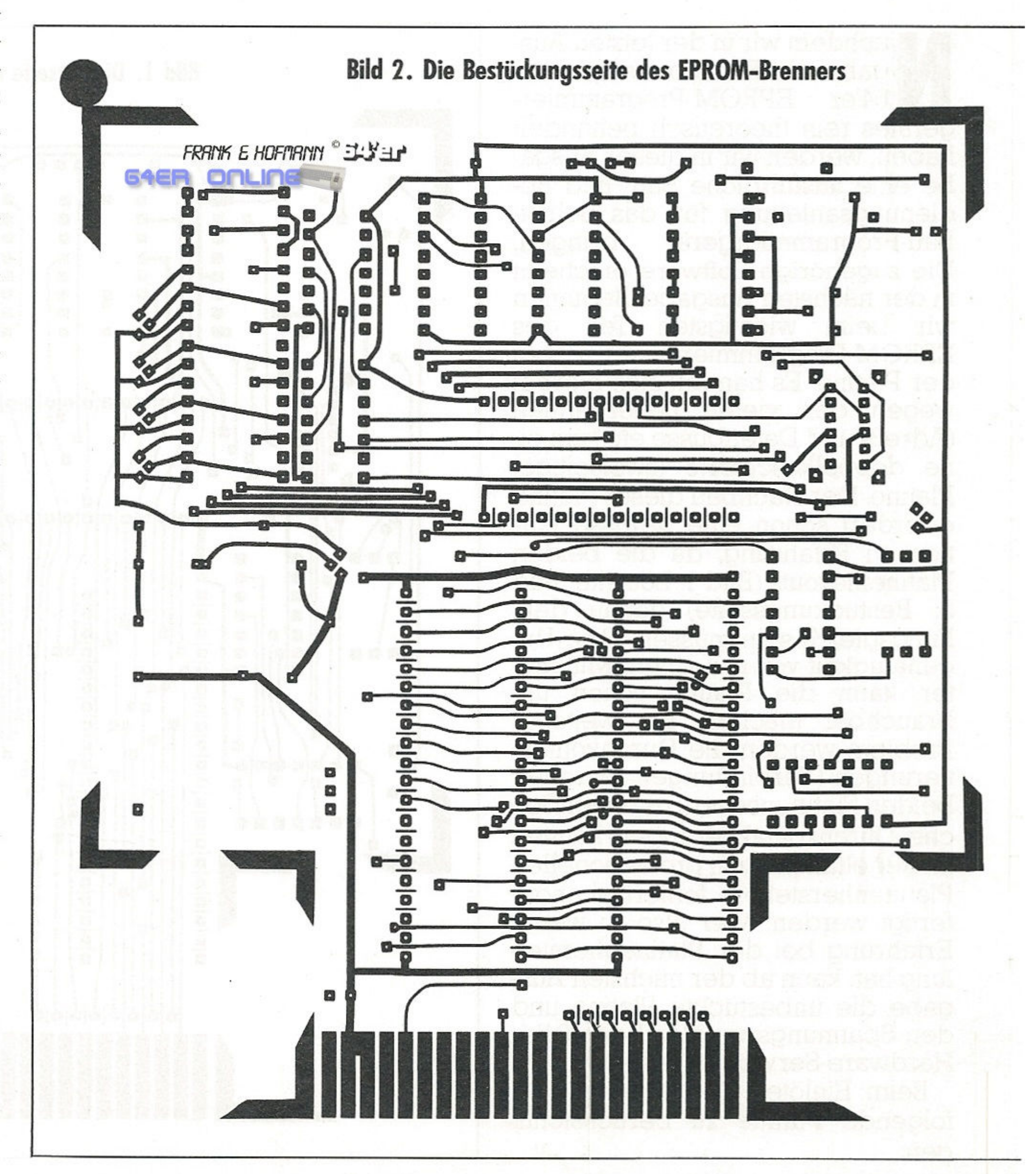


Bild 2. Die Bestückungsseite des EPROM-Brenners

bedingt zu beachten:
1) EPROM immer mit Kerbe in Richtung des Spannhebels einlegen.
2) Beim Einstecken der 24poligen EPROM-Typen die ersten vier Pins (vom Hebel aus gesehen) freilassen.

Achtung: Werden diese beiden Punkte nicht beachtet, so kann das zur Zerstörung des EPROMs führen!!!

Wollen Sie ein EPROM programmieren, so drücken Sie die Taste »P«. Anschließend müssen folgende drei Adressen in hexadezimaler Schreibweise eingegeben werden:
1) Erstes Speicherbyte: Die Adresse des ersten zu programmierenden Bytes im Speicher des Computers.
2) Anzahl der zu programmierenden Bytes.
3) Startbyte im EPROM: Gibt an, ab welcher Adresse im EPROM der Speicherinhalt abgelegt werden soll.

Wurden diese Parameter richtig eingegeben, wird nach Bestätigung der Sicherheitsabfrage mit RETURN, das EPROM programmiert.

Inbetriebnahme des Programmiergerätes

Ein Blinken der Leuchtdiode bestätigt die Programmierung. Der Programmiervorgang dauert relativ lange, da für jedes Byte ein 50 ms langer Programmierimpuls aufgewendet werden muß. Das Programm überprüft außerdem nach jedem gebrannten Byte deren Richtigkeit und gibt gegebenenfalls eine Fehlermeldung aus und bricht den Programmiervorgang ab.

Über den Menüpunkt »Auslesen« kann ein bereits programmiertes EPROM ausgelesen werden. Die Eingaben entsprechen denen der Programmierroutine, nur der Datenfluß ist umgekehrt und natürlich erheblich schneller. Der Leertest überprüft, ob jedes Bit des EPROMs gesetzt ist. Das heißt jede Speicherzelle des EPROMs muß den Wert $FF (dezimal 255) aufweisen.

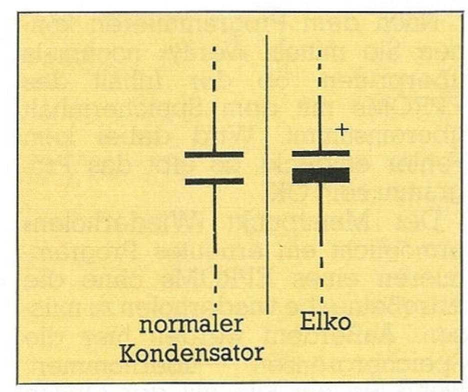


Bild 3. Kondensator und Elko

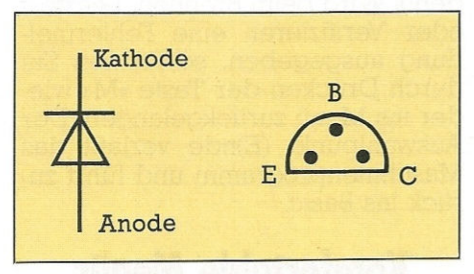


Bild 4. Diode und Transistor

Bild 5. Bauteileliste des EPROM-Brenners

Stückliste für das 64'er EPROM-Programmiergerät

Halbleiter

IC 1/2	6821
IC 3/7	7406
IC 4	7404
IC 5	74 LS 139
IC 6	74 LS 123
IC 8	7408
T 1	BC 337-40
T2 bis T6	BC 327-40
D1-D7	1N4148
ZD 1	ZPD 27
ZD 2/3	ZPD 24
ZD 4/6	ZPD 9.1
ZD 5/7	ZPD 3.3
LD 1	LED 3 mm rot

Kondensatoren

C 1	100 µF/40 V	Elko
C 2	10 µF/40 V	Elko
C 3	6.8 µF/10 V	Tantal
C 4	100 nF/100 V	MKT
C 5/6	10 µF/10 V	Tantal

Widerstände

R 1	1 KΩ	
R 2	3.3 KΩ	4 Stück
R 3	1 KΩ	4 Stück
R 4	20.5 KΩ	Metallfilm 1%
R 5	1 KΩ	2 Stück
R 6	43 KΩ	4 Stück
R 7	4.7 KΩ	
R 8	220 KΩ	
R 9	3.3 KΩ	
R 10	1 KΩ	
R 11	4.7 KΩ	
R 12	10 KΩ	

Sonstige Bauteile

1 Textool Nullkraftsockel 28polig
1 Spannungswandler TDK 05 CE 0072
1 Vierfach-Umschalter Typ SL 422 P
(zum Beispiel Bürklin Nr. 11 G 822 oder Conrad Nr. 70 82 32)

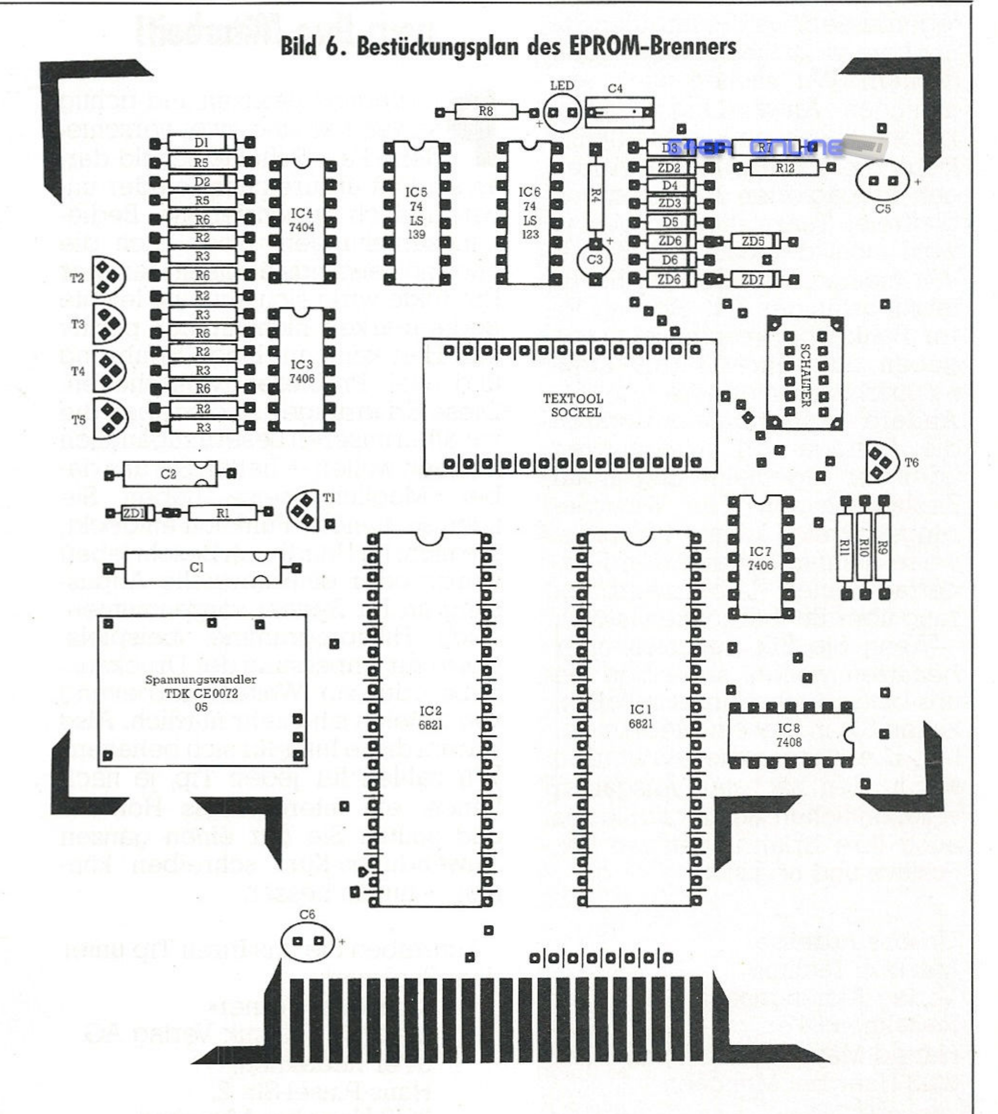


Bild 6. Bestückungsplan des EPROM-Brenners

Nach dem Programmieren können Sie mittels »Verify« nochmals überprüfen, ob der Inhalt des EPROMs mit dem Speicherinhalt übereinstimmt. Wird dabei kein Fehler entdeckt, so gibt das Programm sein OK.

Der Menüpunkt »Wiederholen« ermöglicht ein erneutes Programmieren eines EPROMs ohne die Adreßeingabe wiederholen zu müssen. Außerdem werden hier die Speichergrenzen übernommen, wenn Sie ein File mit der Option (H)ex $1000 laden (siehe File Laden!). Wird beim Brennen, Leertest oder Verifizieren eine Fehlermeldung ausgegeben, so können Sie durch Drücken der Taste »M« wieder ins Menü zurückgelangen. Der Auswahlpunkt (E)nde verläßt das Maschinenprogramm und führt zurück ins Basic.

Komfortable Menüs

Wollen Sie ein Programm von Diskette auf EPROM übertragen, so können Sie dieses mit dem Menüpunkt (F)ile in den Basic-Speicher des C 64 laden. Nach Drücken der »F«-Taste verzweigt das Programm in ein kleines Untermenü, in dem Sie zwei Optionen zum Laden des Files wählen können:

(O)riginaladresse: Das Programm wird an die Originaladresse geladen (zum Beispiel $0801 für Basicprogramme). In diesem Fall muß zum Programmieren des EPROMs der Menüpunkt (P)rogrammieren angewählt werden.

(H)ex $1000: Wählt man diese Option, so wird jedes Programm automatisch nach $1000 geladen und alle Parameter an die Wiederholungsroutine übergeben. In diesem Fall braucht man nur nach dem Laden den Punkt (W)iederholen anwählen und das EPROM wird sofort vollständig programmiert.

Um den Speicherinhalt des EPROMs nicht zu verlieren, empfiehlt es sich, das Sichtfenster am EPROM zuzukleben. Hierfür eignen sich besonders die Schreibschutzaufkleber aus Diskettenpackungen. Dies ist nötig, da sich EPROMs mit UV-Licht löschen lassen. Tageslicht enthält UV-Strahlung.

Im Menü können Sie außerdem durch Betätigung der Funktionstasten F1, F3 und F5 die Farben des Rahmens, des Hintergrundes und der Zeichen beliebig ändern.

Die Software folgt in der nächsten Ausgabe.

(M. Frank/C. Q. Spitzner/og)

Ist Btx nix?

Wenn Sie Erfahrungen mit Btx haben oder an einem Btx-Anschluß Interesse haben, dann schreiben Sie uns bitte. Wir, die 64'er-Redaktion, wüßten gerne, wie gut Btx bei unseren Lesern ankommt.

Haben Sie zu Hause oder im Betrieb einen Btx-Anschluß oder möchten Sie sich einen legen lassen? Dann schreiben Sie uns bitte, wozu Sie Btx verwenden (wollen). Wir suchen nach vernünftigen Anwendungsmöglichkeiten dieses neuen Mediums. Denn die von der Post in Anzeigen propagierten Abfragen von bleifreien Tankstellen etc. sind ja wohl nicht der Sinn der Sache. Wir meinen, das Medium Btx ist leistungsfähiger. Ob Sie nun Ihrer Bank Überweisungsaufträge geben oder Ihrer Firma Kurznachrichten hinterlassen wollen. Andere Anwendungen wären die Abfrage von Datenbanken oder der allgemeine Zugriff auf Rechenanlagen. Ein Versicherungsvertreter könnte beispielsweise seinem Kunden den Endbetrag einer Lebensversicherung über Btx berechnen lassen.

Wenn Sie Btx benutzen oder benutzen wollen, schreiben Sie uns bitte, welche Einsatzmöglichkeiten Sie in diesem Medium sehen. Die interessantesten werden wir in den nächsten Ausgaben veröffentlichen. Schildern Sie uns auch Ihre Erfahrungen mit Btx. Positive und negative.

Unsere Adresse:
Markt & Technik
Verlag Aktiengesellschaft,
Redaktion 64'er,
Harald Meyer, Hans-Pinsel-Str. 2,
8013 Haar bei München

Software-Kenner gesucht!

Kennen Sie sich mit professioneller Software aus, haben Sie Tips zum Umgang mit diesen Programmen, oder wissen Sie, wie man alles aus einem Programm herausholt? Dann sind Sie unser Mann, denn wir suchen für unseren Software-Corner noch Ihre Mitarbeit!

Programme besitzen und richtig anwenden sind zwei verschiedene Paar Schuhe. Schuld daran sind oft unzureichende oder unverständlich geschriebene Bedienungsanleitungen. Aber auch die enorme Leistungsfähigkeit mancher Produkte wirkt sich auf eine leichte Bedienbarkeit nicht immer positiv aus. Hier kann nur lange Erfahrung und viel Probieren weiterhelfen. Diese Erfahrungen sind es aber, die wir allen unseren Lesern zugänglich machen wollen — helfen Sie uns dabei. Möglicherweise haben Sie noch so manche Funktion entdeckt, die nicht im Handbuch beschrieben wurde oder eine spezielle Anpassung an Ihr System vorgenommen. Auch Hilfsprogramme, beispielsweise zur Anpassung der Druckausgabe oder zur Weiterverarbeitung von Dateien sind sehr nützlich. Also warum diese Infos für sich behalten? Wir zahlen für jeden Tip, je nach Länge, ein interessantes Honorar und sollten Sie gar einen ganzen Anwendungs-Kurs schreiben können — um so besser.

Schreiben Sie uns Ihren Tip unter dem Stichwort:
»Software Corner«,
Markt & Technik Verlag AG,
64'er Redaktion,
Hans-Pinsel-Str. 2,
8013 Haar bei München



Fortsetzung von Seite 125

```
0010   IF zeichen$ IN "aeiou" THEN
0020     neu$:="*"
0030   ELIF ORD(zeichen$) > 64 AND ORD(zeichen$) < 91 THEN
0040   neu$:=CHR$(ORD(zeichen$)+128)
0050   ELIF ORD(zeichen$) > 47 AND ORD(zeichen$) < 58 THEN
0060     neu$:=""
0070   ELSE
0080     neu$:zeichen$
0090   ENDIF
```

Listing 22. Textumwandlung in Comal

```
10     rem if elif
20       if not (z$="a" or z$="e" or z$="i" or z$="o" or z$="u")
         then 50
30         neu$="*"
40         goto 120
50       if not (asc(z$) > 64 and asc(z$) < 91) then 80
60         neu$=chr$(asc(z$)+128)
70         goto 120
80       if not (asc(z$) > 47 and asc(z$) < 58) then 110
90         neu$=""
100        goto 120
110      neu$=z$
120    rem endif
```

Listing 23. Textumwandlung in Basic

Das wird häufig vergessen. So heißt es zum Beispiel in letzter Zeit des öfteren, das Basic 7.0 des neuen Commodore 128 mache nun endlich strukturiertes Programmieren möglich, auch in dieser Zeitschrift war solches zu lesen. Gewiß, der erste Schritt wird erleichtert, und das sei dankbar vermerkt. Aber wer ein wirklich »strukturiertes« Basic möchte, der muß sich entweder einen anderen Computer anschaffen, wie zum Beispiel das BBC-Micro, oder eine Basic-Erweiterung, die neben vorgefertigten Steuerbausteinen auch mindestens einige vorgefertigte Unterprogrammbausteine anbietet, wie zum Beispiel das auch in dieser Zeitschrift schon besprochene Macro-Basic.

Um Unterprogrammbausteine, also um Prozeduren und Funktionen, wird es im zweiten Teil dieser Serie gehen.

Zusammenfassung

Fassen wir die Bausteine, die wir kennengelernt haben, noch einmal in einer Übersichtstabelle zusammen (Bild 13).

	Bausteine	
Abfolge oder Sequenz:	**Schleife**	**Verzweigung**
	Zählschleife	Abstecher
	WHILE-Schleife	Gabelung
	UNTIL-Schleife	Mehrfachverzweigung
	LOOP-Schleife	
	Endlosschleife	

Wir haben gesehen, daß diese Bausteine sowohl in einer »strukturierten« Computersprache wie Comal codiert werden können als auch in einer »unstrukturierten« wie Basic. Der Unterschied liegt nur darin, daß Comal für die meisten Bausteine entsprechende Befehlsblocks zur Verfügung stellt, so daß diese Denkschritt für Denkschritt umgesetzt werden können. In Basic hingegen sind wir gezwungen, uns entsprechende Befehlsstrukturen selber zu schaffen. Aber so schwer ist das, wie wir gesehen haben, gar nicht, und wenn wir sie erst einmal haben, dann ist die Bausteinlogik genau so leicht zu codieren wie in Comal.

Entsprechendes gilt für die anderen Bausteintypen, mit denen wir uns noch beschäftigen wollen. (Prof. Burkhard Leuschner/gk)

Hinweise:
Besonders anregend fand ich folgende neuere Literatur:

Kopp, Martin (1984), Neue Strukturen im alten Basic, INFO (Rundbrief des PTC, Offenburg) 6(1), 57-58
Lührmann, Arthur (1983), Slicing through spagetti code, The Computing Teacher 10(8), 9-15
Lührmann, Arthur (1984), Structured Programming in Basic, Creative Computing 10(5), 152-156; 10(6), 152-163; 10(7), 125-136; 10(9), 171-177
Metzler, Richard C.(1985), If rules then better structured Basic, The Computer Teacher 12(4), 12-14

Wenn Sie Comal unerwarteterweise noch nicht haben — die Diskettenversion 0.14 (C 64, CBM 40332/8032) ist gegen geringe Unkosten zu erhalten. Andere Versionen (auch für andere Computer, zum Beispiel IBM-PC) kosten Geld. Wenn Sie Näheres wissen wollen: Adressieren Sie einen Umschlag, schreiben Sie »Drucksache« drauf und frankieren Sie ihn mit 50 Pfennig. Schicken Sie diesen Umschlag an: »Professsor Burkhard Leuschner, Kennwort Comal, Pädagogische Hochschule, Oberbettringer Str. 200, 7070 Schwäbisch Gmünd«. Sie erhalten dann die neuesten Bezugsinformationen. Bitte keine Mitteilungen beilegen, sie werden nicht gelesen! Und ein paar Tage Geduld haben!

Softlearning — Lernen auf ganz neue Art?

Softlearning erschließt eine neue Methode des Lernens, das »Superlearning«. Wie erfolgreich ist diese neue Art des Lernens?

Über diese neue Art des Lernens, das auf psychologischen und biochemischen Vorgängen beruht, wurde in der Fachwelt viel diskutiert. Das Buch »Superlearning« der beiden Autoren Ostrander und Schröder kann als Vorbild für das Computerprogramm von SM Software angesehen werden. Mit der neuen »Amadeus«-Reihe wurde das bisherige Softlearning um eine neue Variante bereichert.

Schon vor der Geburt im Mutterleib lernt der Mensch ununterbrochen. Dieses für das Überleben so wichtige Lernen geschieht freiwillig und automatisch. Wesentliche Faktoren sind neben dem Inhalt aber auch die Intensität und die Bereitschaft Information aufzunehmen.

Beispielsweise lernt man als Kind deshalb so schnell und leicht die erste Sprache, da man mit dieser seine Wünsche wesentlich besser artikulieren und in Kommunikation mit seinen Mitmenschen treten kann. In der Schule verliert man im allgemeinen diese positive Einstellung zum Lernen, man muß sich dazu zwingen und ungeliebten Stoff in sich hineinpauken. Das wird um so anstrengender und weniger effektiv, je weniger man den Sinn dieses Stoffes versteht. Von diesen Erkenntnissen ausgehend, versuchte man neue Formen des Lernens zu finden. Ergebnis ist das sogenannte »Superlearning«, das in einer Tiefenentspannungsphase Information direkt ins Unterbewußtsein bringt. Auf diese Art verabreichtes Wissen sitzt dann wesentlich fester im Gedächtnis und benötigt wesentlich weniger Zeit zur Vermittlung. Man versucht dabei Lernstoff an der Bewußtseinsschwelle vorbei direkt in das Lang-



Multiple Choice bei SM-Softlearning

Fortsetzung auf Seite 157



Bestellungen aus anderen Ländern bitte per Auslandspostanweisung!

Bestellungen aus der Schweiz bitte direkt an: Markt&Technik Vertriebs AG Kollerstrasse 3 CH-6300 Zug Tel. 042/41 56 56

Bestellungen aus Österreich bitte direkt an: Bücherzentrum Meidling Schönbrunnerstr. 261 1120 Wien Tel. 0222/833196 Microcomput-ique Erhard Schiller Fasangasse 21 1030 Wien Tel. 0222/785661

Hardware für alle - ein neuer 64'er Leser-Service

Der Commodore 64 hat schon oft bewiesen, wie vielseitig er ist. Er läßt sich nicht nur mit Programmen, sondern auch durch so manche Hardware-Erweiterung sinnvoll nutzen und ausbauen. Dabei ist es sicherlich ein reizvoller Bestandteil des Computer-Hobbys, sich solche Erweiterungen selbst nachzubauen. Aber nicht jeder Leser verfügt über die Gelegenheit und Zeit zur Platinenherstellung. Hinzu kommt, daß es oft zu teuer ist, wegen einer bestimmten Erweiterung, Investitionen von mehreren hundert Mark für eine Platinenstation zu tätigen. Wir haben reagiert: Ab sofort besteht die Möglichkeit, im Rahmen des Leser-Service, die in der 64'er abgedruckten Hardware-Erweiterungen in drei verschiedenen Ausbaustufen zu erhalten:

1. Als Platinen
Nur Leerplatinen. Die Beschaffung der Bauteile und der Zusammenbau bleibt bei Ihnen.

2. Als Bausätze
Unsere Bausätze enthalten alle Teile, die notwendig sind, um die beschriebene Erweiterung komplett aufzubauen. Sie brauchen die Bauteile nur noch gemäß der Anleitung in dem jeweiligen Heft zusammenzulöten und einzubauen.

3. Als Fertiggeräte
Die Fertiggeräte sind komplett aufgebaute und geprüfte Geräte. Sie brauchen die Erweiterung lediglich noch einzubauen.

Wichtiger Hinweis: Wir bemühen uns um eine umgehende Auslieferung Ihrer bestellten Hardware. Aber bis zum Eingang Ihrer Überweisung, der Auftragsabwicklung und der dazugehörenden Postwege vergehen mindestens 3 Wochen. Bitte haben Sie Verständnis, wenn aus diesen Gründen Ihre Hardware nicht sofort bei Ihnen eintrifft.

Unser Angebot

Angebot 1:

Expansion-Port Eprom-Platine mit 1 x 8 KByte Speicherplatz für 2732 bis 2764 Eproms.

Beschreibung in Ausgabe 10/85
Bestellnummer: HW 010 pro Stück **19,80***
Dieser Artikel wird nur als Fertiggerät angeboten.

Angebot 2:

Expansion-Port Eprom-Platine mit 2 x 8 KByte Speicherplatz für 2732 bis 2764 Eproms, mit Umschaltmöglichkeit.

Beschreibung in Ausgabe 10/85

Leerplatine
Bestellnummer: HW 020 pro Stück **24,80***

Bausatz mit allen Teilen:
Bestellnummer: HW 021 pro Stück **49,80***

Fertiggerät, getestet, wie beschrieben:
Bestellnummer: HW 022 pro Stück **59,80***

Angebot 3:

Eprom Trans – Die Speichererweiterung

ROM-Speichererweiterung zum Einbau in den C64, gleichzeitig Steckplatz für ein Original- oder ein alternatives Betriebssystem. Zwei Platinen in Epoxid-Harz-Ausführung wie in Ausgabe 10/85 beschrieben.

Leerplatine
Bestellnummer: HW 030 pro Stück **49,80***

Bausatz mit allen Teilen:
Bestellnummer: HW 031 pro Stück **119,80***

Eprom-Trans ist nicht als Fertiggerät erhältlich. Die Hardware-Erweiterungen aus früheren Ausgaben und die 40/80 Zeichen-Umschaltung für den C128 werden wir so bald als möglich in unser Angebot aufnehmen.

Angebot 4:

Super Kernal

Erweitertes Betriebssystem für den C 64 mit vielen neuen Funktionen inkl. Adaptersockel, einbaufertig in den C 64.
Beschreibung in Ausgabe 11/85

Version 1: Enthält Hypra Load / DOS 5.1 / Funktionstastenbelegung / Renew / RS232
Bestellnummer: HW 040

Version 2: Enthält Hypra Load / DOS 5.1 / Funktionstastenbelegung / Renew / Super Centronics Schnittstelle
Bestellnummer: HW 041

Version 3: Enthält Hypra Load / DOS 5.1 / Funktionstastenbelegung / Renew / Hypra Save
Bestellnummer: HW 042

Version 4: Enthält Hypra Load / DOS 5.1 / Funktionstasten / Hypra Save / Centronics klein
Bestellnummer: HW 043

Preis für jede Version pro Stück: **39,80***

* Alle Preise inklusive Mehrwertsteuer

Qualität & Service

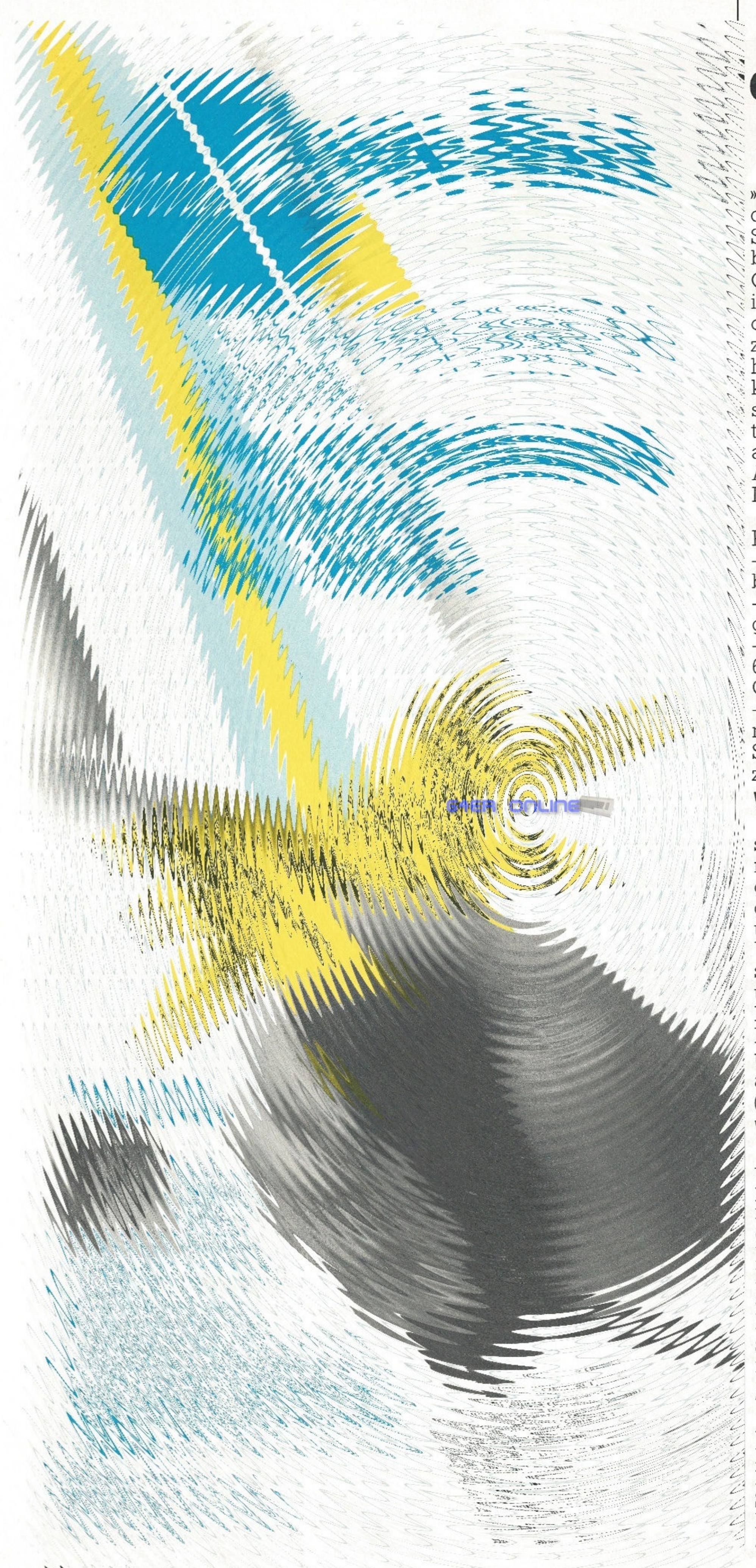
- **Die 64'er Hardware hat einen hohen Qualitätsstandard. Wir verwenden nur beste Epoxid-Harz-Platinen mit Lötstopp-Lack.**
- **Wir verwenden nur Präzisionssockel mit gedrehten Kontakten.**
- **Alle Platinen werden professionell gefertigt. Wenn notwendig mit doppelseitiger Beschichtung und Löt-Durchkontaktierungen.**
- **Jedes Gerät, das wir versenden, wurde auf Funktionstüchtigkeit geprüft.**
- **Wir sind auch nach dem Verkauf für Sie da.
Neben der gesetzlichen Garantie bietet unser Service- und Fertigungspartner Ihnen Hilfe und Unterstützung an.**

Unsere Garantie

Im Rahmen der Versand- und Lieferbedingungen unterliegen die Geräte einer Gewährleistungszeit von 6 Monaten ab Lieferung. Der Lieferung liegt eine Service-Karte bei, die Sie im Falle einer Beanstandung zusammen mit dem Gerät an die auf der Karte vermerkte Adresse schicken können. Die gleiche Karte verwenden Sie bitte bei Reparaturen nach der Garantiezeit.

Wie bestelle ich?

Alle Hardware-Erweiterungen, die Sie bestellen können, tragen einen Bestellverweis am Ende des Artikels im jeweiligen Heft. Falls Sie keinen Hinweis finden, hat sich der Autor dieser Erweiterung nicht dazu entschließen können, seine Entwicklung im Rahmen des Leserservice für eine Verbreitung freizugeben. Bitte verwenden Sie für Ihre Bestellung immer die beiliegende Postscheck-Zahlkarte oder einen Verrechnungsscheck. Sie erleichtern uns damit die Auftragsabwicklung und sparen sich Versandkosten.



Gesucht: Software für C 16/C 116/Plus 4

Die Commodore-Computer der »C 16-Generation« stehen und standen ja leider von Anfang an stark im Schatten des »Superstars« C 64. Dabei weisen gerade diese Computer Qualitäten auf, die sie eigentlich zu idealen »Einsteiger«-Maschinen machen sollten: Ein wirklich ausgezeichnetes und komfortables Basic, hochauflösende Grafik und ein kaum noch zu unterbietender Preis sind die Pluspunkte dieser Computer. Nachteilig ist in erster Linie das arg begrenzte Software-Angebot. Aber dem läßt sich ja abhelfen — mit Ihrer Hilfe:

Wenn Sie für den C 16, C 116 oder Plus 4
— ein gutes Spiel geschrieben haben,
— ein brauchbares Anwenderprogramm in der Schublade haben,
— VC 20/C 64-Programme aus dem 64'er-Magazin oder aus Happy-Computer umgeschrieben haben,
— Tips und Tricks kennen, die noch nicht gedruckt wurden, dann sollten Sie uns Ihr Programm oder Ihren Tip zuschicken.

Was wir von Ihnen brauchen

Wenn Sie uns ein Programm einsenden, dann brauchen wir von Ihnen auf jeden Fall einen Datenträger (Diskette, Kassette) mit dem Programm, eine möglichst ausführliche und informative Beschreibung. Von Vorteil ist die Mitlieferung von Listing, Variablen- und Programmübersicht sowie gegebenenfalls von Hardcopies und Beispielausdrucken. Bitte geben Sie im Anschreiben unbedingt Computertyp, Speicherausbau und verwendete Peripherie (Floppy, Drucker etc.) an.

Was mit Ihrer Einsendung geschieht

Ihr Programm wird in der Redaktion getestet und auf Eignung überprüft. Falls es zum Abdruck geeignet ist, erhalten Sie innerhalb weniger Wochen einen entsprechenden Bescheid, andernfalls gehen Ihre kompletten Unterlagen automatisch an Sie zurück. Bei Abdruck Ihres Programms erhalten Sie selbstverständlich ein angemessenes Honorar.

An wen Sie Ihr Programm einsenden

Schicken Sie Ihre kompletten Unterlagen an

Markt & Technik Verlag AG
Redaktion 64'er
Stichwort: Aktion C 16
Hans-Pinsel-Str. 2
8013 Haar bei München

Fortsetzung von Seite 154

zeitgedächtnis zu bringen. Das Softlearning-Programm nutzt diese Methode und verbindet dabei mehrere moderne Hilfsmittel wie Computer und Kassettenrecorder. Der Computer ist deshalb so gut für diese Art des Lernens geeignet, da er ein nie nachlassender Lernpartner ist.

Die Komponenten

Neben einer Computeranlage, bestehend aus C 64, Floppy 1541 und Bildschirm, braucht man für Softlearning noch einen Recorder, der mit Audio- und Fernbedienungsbuchse ausgestattet sein muß. Die Audiobuchse wird zur Synchronisation zwischen Recorder und Computer gebracht, während der C 64 das Kassettengerät mit Hilfe der Fernbedienungsbuchse steuern kann. Die Audiobuchse muß dabei unbedingt vorhanden sein, während das Anhalten und Weiterlaufenlassen auch per Hand erledigt werden kann. Weiterhin sind dazu noch die Systembasis, bestehend aus Systemdiskette, Audioadapter und Handbuch nötig. Diese Systembasis ist die Grundlage für alle angebotenen Softlearning-Kurse, also auch der Amadeus-Reihe. Als letzter Teil dieser ganzen Ausstattung kommt ferner noch der eigentliche Sprachkurs hinzu. Es werden hier inzwischen Grund-, Aufbau- und Intensivkurse für die Sprachen Englisch, Französisch, Italienisch und Spanisch angeboten.

Der Aufbau

Zunächst einmal sollte man sich mit Hilfe des angebotenen Lockerungstrainings in einen Entspannungszustand, auch Alpha-Zustand genannt, bringen. Diese Phase ist für das Lernen nach dieser Methode entscheidend. Diese Übung sollte man am besten mehrfach vor dem eigentlichen Einstieg in den Sprachkurs durchführen, da man den Alpha-Zustand künftig am Anfang jeder Übung des Sprachkurses erreichen muß. Nun werden in gelockerter Atmosphäre Dialogtexte in der neuen Sprache vorgesprochen und auf dem Bildschirm synchron dargestellt. Diese Texte werden bei richtiger Anwendung vom Unterbewußtsein aufgenommen und gespeichert. Nach Ablauf dieser Texte wiederholt man den gesamten Vorgang zur Vertiefung noch einmal. Jetzt folgt das eigentliche Training mit den in den Dialogtexten verwendeten Wörtern, damit das erlernte Wissen im Gedächtnis aktiviert wird. Das Training besteht aus mehreren Übungen, in denen die einzelnen Wörter nach unterschiedlichen Kriterien abgefragt und entsprechend Punkte vergeben werden. Dabei werden Wörter, die noch nicht so fest im Gedächtnis haften und bei denen man deshalb mehr Fehler macht, öfter abgefragt. An Übungen gibt es da beispielsweise das Multiple Choice-Verfahren (siehe Bild), bei dem unter mehreren angegebenen Bedeutungen die richtige herausgesucht werden muß. Weiterhin muß man im Silbenrätsel einzelne Wortsilben zur richtigen Bedeutung des gefragten Wortes zusammensetzen. Altbekannt ist der Zettelkasten, bei dem man die richtige Bedeutung des angezeigten Wortes angeben muß.

Bei dieser Arbeit helfen »Keywords«. Das sind Wörter, die dem fremdsprachigen Wort so ähnlich wie möglich klingen und damit quasi eine Art »Eselsbrücke« aufbauen. Die Verwendung dieser Keywords erleichtert das Erlernen schwieriger Wörter.

Um Klang und Bild einer Sprache besser verbinden zu können, wird ein Lückentextdiktat verwendet. Dazu wird ein am Anfang in der Tiefenentspannungsphase vorgesprochener Text benutzt. Dieser Text enthält Lücken, die zu schließen sind. Dabei bekommt man auch wieder Punkte, die zum Punktekonto hinzuaddiert werden. Ähnlich verläuft die Grammatikübung, bei der man auch wieder Lücken in vorgegebenen Text ausfüllen muß. Es werden bei den Grund- und Aufbaukursen etwa 1200 bis 1500 Wörter vermittelt. Bei den halb so langen Intensivkursen sind es ungefähr 800 Wörter. Diesen Wortschatz kann man mit zusätzlichen Kassetten um jeweils 1000 Vokabeln erweitern. Weiterhin angeboten wird ein Editor, mit dem man selbst den zu trainierenden Wortschatz ausbauen kann.

Gutes Konzept

Das System des Superlearning wird seit längerer Zeit vor allem in der Managerebene mit gutem Erfolg angewendet. Es ist damit möglich, sich neues Wissen sehr effektiv anzueignen. Entscheidend für den Erfolg mit dieser Methode ist aber der nötige Ernst bei der Sache und die Bereitschaft, das System mit aller Ruhe anzuwenden. Man muß zuerst daran glauben, daß diese Methode funktioniert, sonst erreicht man nie die so wichtige Tiefenentspannungsphase beim Durcharbeiten der einzelnen Übungsteile. Leider ist die Umsetzung dieser neuen Lernmethode auf den C 64 und Kassettenrecorder bei Softlearning nicht optimal gelungen. So stören teilweise erhebliche Wartezeiten beim Nachladen einzelner Programmteile und Informationen den Ablauf der Lektionen und auch die beim C 64 so gefürchtete Garbage Collection tut ein übriges. Dies wäre bei geeigneter Programmierung sicher vermeidbar gewesen. Weiterhin funktioniert die Synchronisation zwischen Computer und Recorder manchmal nicht einwandfrei, so daß der Dialogablauf in der Tiefenentspannungsphase aus dem Takt gerät.

Weiterhin ist Softlearning nicht gerade billig. So kostet die unbedingt nötige Systembasis »S« 89 Mark und die eigentlichen Sprachkurse (Grundkurs) kommen noch einmal auf je 198 Mark. Für die Intensivkurse der Amadeus-Reihe muß man pro Kurs 98 Mark ausgeben. Außerdem kann man einen Zusatzwortschatz für 49 Mark erwerben. Wer sich seinen eigenen Wortschatz erstellen möchte, kann sich für 49 Mark noch einen Wortschatzeditor zulegen. Wenn man also eine Sprache umfassend erlernen will, sind die dazu notwendigen Ausgaben durchaus nicht zu verachten.

Angeboten werden zur Zeit Grundkurse in Englisch, Französisch, Spanisch und Italienisch. Intensivkurse gibt es für Englisch, Französisch, Italienisch, Schwedisch, Spanisch und Russisch. Für Industriekapitäne ist auch noch ein spezieller Kurs in Management-Englisch erhältlich, von dem es aber noch keinen Intensivkurs gibt.

Fazit

Abschließend kann man sagen, daß diese neue Lernmethode bei Softlearning im großen und ganzen zufriedenstellend auf die Homecomputerebene umgesetzt worden ist. Man bekommt die Möglichkeit geboten, eine neue Sprache in ungewöhnlicher Technik zu erlernen. Wenn man dabei ernsthaft und bereitwillig die gebotenen Möglichkeiten ausnutzt und anwendet, kann man in kürzester Zeit zu ausgezeichneten Ergebnissen kommen.

(Karl Hinsch/aw)

Info: SM-Soft-Training, Fasanengartenstraße 4, 8000 München 83

Spieletests: So wird bewertet

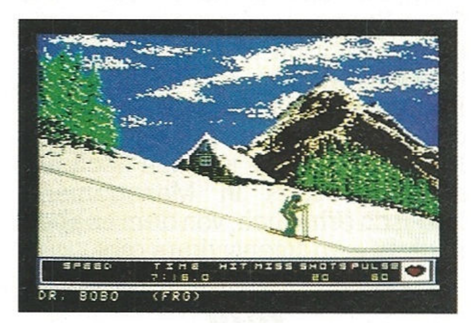
In letzter Zeit erreichten uns immer mehr Anfragen nach den komischen schwarzen Balken unter den Spieletests. Deshalb erläutern wir mal kurz Konzept und Bewertungsschema unseres Spieleteils.

Wie Sie sicherlich bemerkt haben, hat sich das Gesicht unserer Spieletests etwas gewandelt. Geblieben sind allerdings die Bewertungsbalken, die sich nach Ansicht unserer Leser sehr bewährt haben. Sie verlangten aber nach noch besseren Vergleichsmöglichkeiten bei den Spielen. Diesen Wunsch haben wir uns bei der Neukonzeption des Spieleteils zu Herzen genommen. Wir suchen uns immer mehrere Spiele vom gleichen Typ oder gleicher Konzeption heraus und stellen sie auf einer Seite gegenüber. Aber eines wird beim Spieleteil immer berücksichtigt: Jedes Spiel, das wir Ihnen vorstellen, ist gut und sein Geld wert. Denn für schlechte Spiele wollen wir unseren Platz nicht vergeuden.

Nun zu den einzelnen Balken in den Bewertungskästen. In den fünf Kategorien vergeben wir zwischen 0 und 15 Punkten. Da wir sowieso nur gute Spiele testen, fangen die Skalen im Heft erst bei fünf Punkten an. Mit dem Begriff »Spielidee« bezeichnen wir die Handlung des Spiels, seine Originalität, die Grundidee und die Hintergrund-story. »Grafik« und »Sound« brauchen wohl nicht näher erläutert zu werden. Um Vergleichsmöglichkeiten zu schaffen, haben wir im Juli 1985 zwei Maßstabsspiele gesetzt, die jeweils elf Punkte in diesen Kategorien erhalten haben: »Impossible Mission« in puncto Grafik und »Ghostbusters« in puncto »Sound«. Die beste Grafiknote bisher hatte »Summer Games II« mit 13 Punkten, die beste Sound-Note ging mit 12 Punkten an »Thing on a String« und »Rock'n'Bolt«. Mit dem Schwierigkeitsgrad tun wir uns meistens am schwersten. Wir versuchen immer, einen möglichst fairen Durchschnittswert zu ermitteln, der die Schwierigkeit am besten wiedergibt. Zu diesem Punkt finden Sie öfters auch Anmerkungen im Kasten. Der letzte Punkt, Motivation, ist sowohl eine Durchschnittsnote aus den vorherigen Werten wie auch eine abschließende Gesamtkritik. Wie oft, wie lange kann man das Spiel spielen und Spaß dabei haben? Legt man es nach zwei Tagen in die Ecke oder kommt man wochenlang nicht vor drei Uhr früh ins Bett?

Dieses Konzept kam bei unseren Lesern gut an und soll bis auf weiteres auch beibehalten werden. Das heißt natürlich nicht, daß wir uns allen Verbesserungsvorschlägen verschließen. Wenn Sie also was zu meckern, aber auch zu loben haben, dann schreiben Sie uns doch. In diesem Sinne: Viel Spaß beim Spielen! (bs)

Olympia eiskalt



Seit der Vorankündigung in »Summer Games II« fiebern die Sportspiel-Fans der Fortsetzung »Winter Games« entgegen. Jetzt ist es lieferbar.

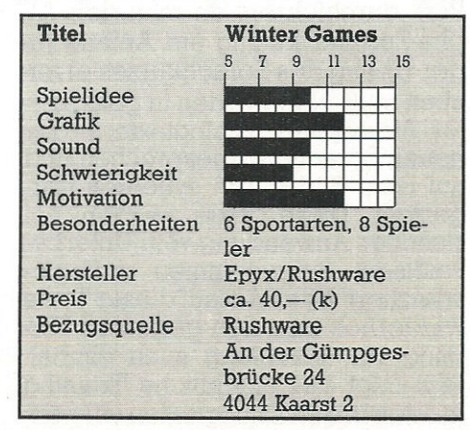
Die Olympiade ist komplett, denn Epyx veröffentlichte mit »Winter Games« das dritte Spiel der Olympia-Trilogie. Und eines gleich vorweg: Noch ein viertes Spiel würde wohl keinen Erfolg mehr haben, denn das bekannte Spielprinzip der Bildschirm-Olympiade zeigt bei »Winter Games« die ersten Ermüdungserscheinungen.

Im bewährten Menü werden den bis zu acht Spielern sieben Disziplinen angeboten, wobei der Eiskunstlauf aber zweimal auftaucht, einmal als Pflicht und einmal als Kür. Die anderen Disziplinen sind Ski-Kunstsprung (Hot Dog), Ski-Weitsprung, Biathlon, Eisschnellauf und Bobfahren.

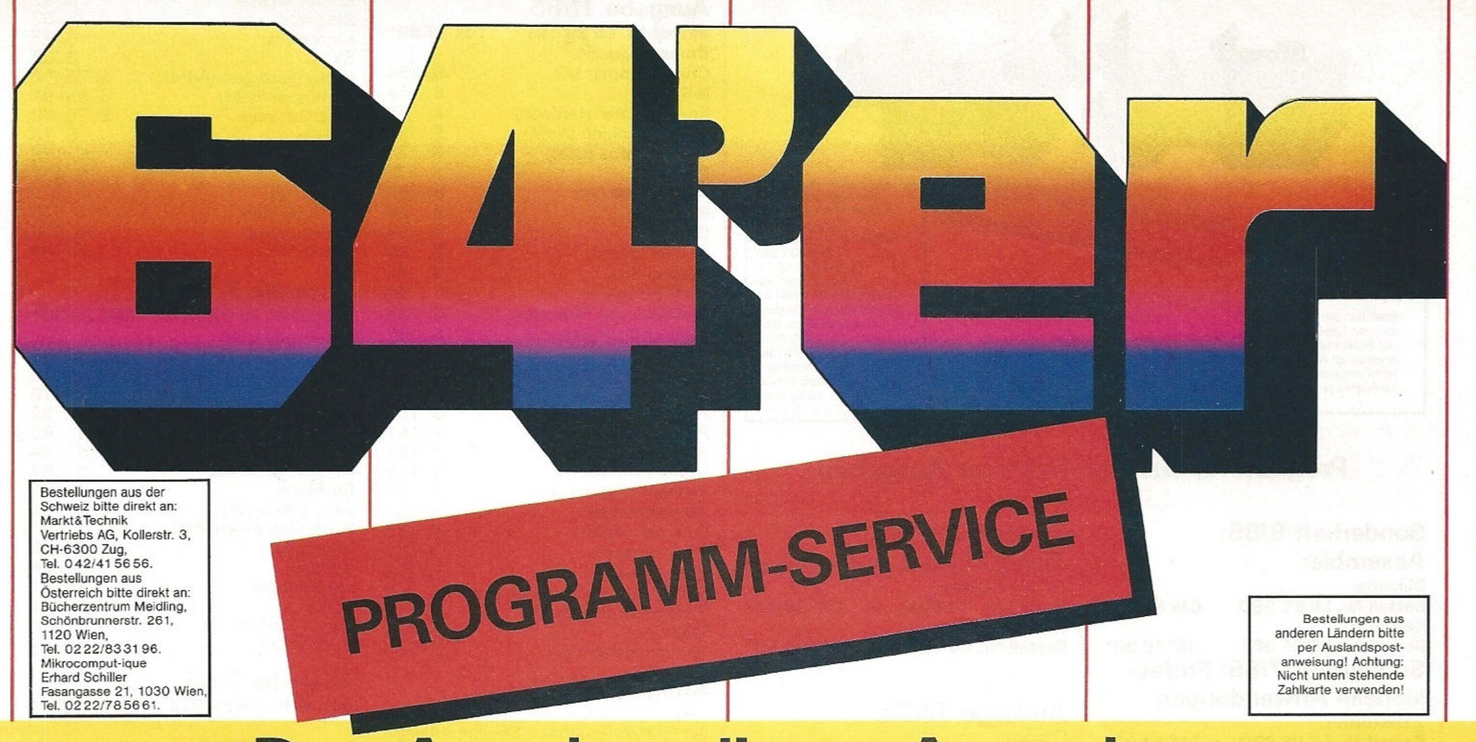
Im einzelnen streuen die Disziplinen sehr in ihrer Qualität. Eindeutig am schlechtesten sind beide Kunstlauf-Disziplinen, die grafisch eher dem ersten Teil von »Summer Games« entsprechen. Ebenso der Eisschnellauf, der eine Kopie des Ruderns aus »Summer Games II« ist, bei der nur die Sprites geändert werden.

Der Kunstsprung erinnert stark an das Turmspringen aus »Summer Games«, kann aber durch Variantenreichtum und bessere Animation überzeugen. Auch das Bobfahren bietet keine großen Sensationen, sondern ist guter »Summer Games II«-Durchschnitt. Fantastisch gelungen sind aber Biathlon und Weitsprung. Tolle neue Ideen und sehr gute Ausführung in Bild wie Ton lassen das Herz aller Spieler höherschlagen. Allein diese beiden Disziplinen rechtfertigen den Kauf von »Winter Games«.

Negativ aufgefallen ist uns, daß die »Summer Games«-Disziplinen nicht in »Winter Games« eingebunden werden können, wie es bei »Summer Games II« der Fall ist. Trotzdem darf man »Winter Games« als würdigen Abschluß der Olympia-Spiele bezeichnen. (bs)



Titel	Winter Games
	5 7 9 11 13 15
Spielidee	
Grafik	
Sound	
Schwierigkeit	
Motivation	
Besonderheiten	6 Sportarten, 8 Spieler
Hersteller	Epyx/Rushware
Preis	ca. 40,— (k)
Bezugsquelle	Rushware An der Gümpgesbrücke 24 4044 Kaarst 2



Bestellungen aus der Schweiz bitte direkt an: Markt&Technik Vertriebs AG, Kollerstr. 3, CH-6300 Zug, Tel. 042/41 56 56.
Bestellungen aus Österreich bitte direkt an: Bücherzentrum Meidling, Schönbrunnerstr. 261, 1120 Wien, Tel. 0222/83 31 96.
Mikrocomput-ique Erhard Schiller Fasangasse 21, 1030 Wien, Tel. 0222/78 56 61.

Bestellungen aus anderen Ländern bitte per Auslandspostanweisung! Achtung: Nicht unten stehende Zahlkarte verwenden!

Das Angebot dieser Ausgabe:

Wer keine Zeit oder keine Lust hat, alle Programme selbst in mühevoller Kleinarbeit abzuschreiben, kann wieder auf den bewährten Diskettenservice zurückgreifen. Alle Programme, die mit dem Diskettensymbol im Inhaltsverzeichnis gekennzeichnet sind, gibt's auf Diskette.

Der Diskette liegen keinerlei Informationen bei. Lesen Sie aufmerksam die Anleitung (ob SYS-Befehle nötig sind, in welcher Reihenfolge geladen werden muß, eventuelle Sprach- oder Speichererweiterungen und ähnliches mehr) in dem jeweiligen Artikel nach. Aus Aktualitätsgründen wird jeweils die abgedruckte Version angeboten. **Eventuelle systematische Fehler, die sich noch im Programm befinden können, müssen von Ihnen selbst, nach Studium des Druckfehlerteufelchens, korrigiert werden. Der detaillierte Disketteninhalt wird mit den Seitenzahlen in der nächsten Ausgabe abgedruckt.**

Wenn Sie Fragen zu den im Programm-Service angebotenen Programmen haben, wenden Sie sich an das Redaktionsteam vom 64er Magazin. Tel. 089/46 13-202.

Bestell-Nr. L6 86 01D (Diskette) DM 29,90*

* inkl. Mehrwertsteuer

Bitte verwenden Sie für Ihre Bestellung **nur** die abgedruckte Postscheck-Zahlkarte zur Überweisung des Rechnungsbetrages oder senden Sie uns einen Verrechnungsscheck. Damit erleichtern Sie uns die Auftragsabwicklung und Sie **sparen** sich die Versandkosten!



Einlieferungsschein/Lastschriftzettel
(nicht zu Mitteilungen an den Empfänger benutzen)

Gebühr für die Zahlkarte
(wird bei der Einlieferung bar erhoben)

bis 10 DM — — — 90 Pf
über 10 DM (unbeschränkt) 1,50 DM
Bei Verwendung als Postüberweisung gebührenfrei

Bedienen Sie sich der Vorteile eines eigenen Postgirokontos
Auskunft hierüber erteilt jedes Postamt

Feld für postdienstliche Zwecke

Hinweis für Postgirokontoinhaber:
Dieses Formblatt können Sie auch als Postüberweisung benutzen, wenn Sie die stark umrandeten Felder zusätzlich ausfüllen. Die Wiederholung des Betrages in Buchstaben ist dann nicht erforderlich. Ihren Absender (mit Postleitzahl) brauchen Sie nur auf dem linken Abschnitt anzugeben.
1. Abkürzung für den Namen Ihres Postgiroamts (PGiroA) siehe unten
2. Im Feld »Postgiroteilnehmer« genügt Ihre Namensangabe
3. Die Unterschrift muß mit der beim Postgiroamt hinterlegten Unterschriftsprobe übereinstimmen
4. Bei Einsendung an das Postgiroamt bitte den Lastschriftzettel nach hinten umschlagen

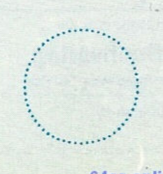
Abkürzungen für die Ortsnamen der PGiroÄ:

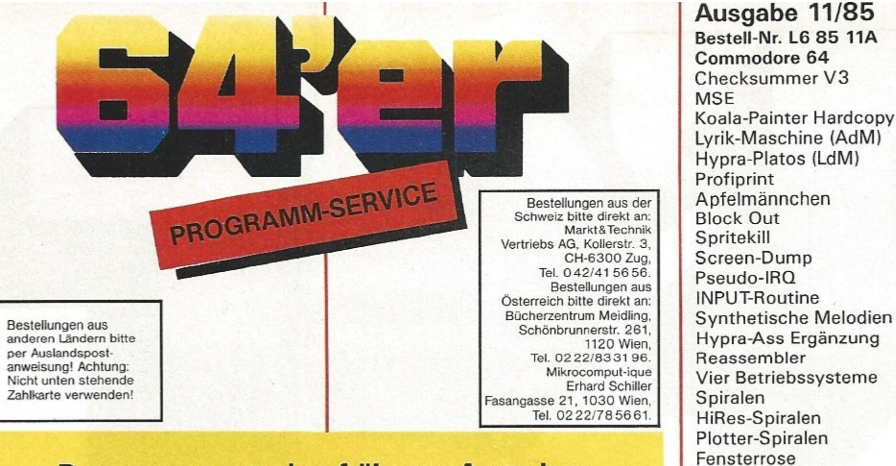
Bln W = Berlin West	Kln = Köln
Dtmd = Dortmund	Lshfn = Ludwigshafen am Rhein
Esn = Essen	Mchn = München
Ffm = Frankfurt am Main	Nbg = Nürnberg
Hmb = Hamburg	Sbr = Saarbrücken
Han = Hannover	Stgt = Stuttgart
Klrh = Karlsruhe	

Für Mitteilungen an den Empfänger

Bestellung Programm-/Hardware -Service Wichtig: Lieferanschrift (Rückseite) nicht vergessen!

Bestell-Nr.	Anzahl	x Einzelpreis	= Gesamtpreis
Summe bitte auf Vorderseite übertragen		Gesamtsumme:	





Bestellungen aus der Schweiz bitte direkt an: Markt&Technik Vertriebs AG, Kollerstr. 3, CH-6300 Zug, Tel. 042/41 56 56. Bestellungen aus Österreich bitte direkt an: Bücherzentrum Meidling, Schönbrunnerstr. 261, 1120 Wien, Tel. 02 22/83 31 96. Mikrocomput-ique Erhard Schiller Fasangasse 21, 1030 Wien, Tel. 02 22/78 56 61.

Bestellungen aus anderen Ländern bitte per Auslandspost-anweisung! Achtung: Nicht unten stehende Zahlkarte verwenden!

Programme aus den früheren Ausgaben

Sonderheft 8/85: Assembler

Diskette
Bestell-Nr. L6 85 S8D DM 29,90*
Kassette
Bestell-Nr. L6 85 S8K DM 19,90*

Sonderheft 7/85: Professionelle Anwendungen

2 Disketten
Bestell-Nr. L6 85 S7D DM 34,90*
4 Kassetten
Bestell-Nr. L6 85 S7K DM 34,90*

Sonderheft 6/85: Top-Themen

2 Disketten
Bestell-Nr. L6 85 S6 DM 34,90*

Sonderheft 5/85: Floppy, Datasette

Diskette
Bestell-Nr. L6 85 S5D DM 29,90*
Kassette
Bestell-Nr. L6 85 S5K DM 19,90*

Sonderheft 4/85: Grafik

Bestell-Nr. L6 85 S4A DM 29,90*

Sonderheft 3/85: Spiele

Beide Disketten in einem Paket!
Verwenden Sie **nur** diese Bestell-Nr.:
Bestell-Nr. L6 85 S3 A DM 34,90*

Sonderheft 2/85: Abenteuerspiele

Bestell-Nr. L6 85 S2 DM 34,90*

Sonderheft 1/85: Tips & Tricks (2. überarb. Auflage)

Floppy-Utilities
Bestell-Nr. CB 023 DM 29,90*
Hilfsprogramme
Bestell-Nr. CB 024 DM 29,90*

Ausgabe 12/85

Diskette
Bestell-Nr. L6 85 12D DM 29,90*
Kassette
Bestell-Nr. L6 85 12K DM 29,90*

Programm	Seite
Checksummer V3	S. 54
MSE V1 .0	S. 54
Old für C 128	S. 43
Chemie-Assistent	S. 57
SMU	S. 68
Hyperscreen	S. 76
Grafik-80	S. 80
Seeschlacht	S. 93
Eprom-Automat	S. 93
Tipp-Utility	S. 99
Floppymonitor	S. 105
Auto.OBJ	S. 108
Bildsch.Langsam	S. 107
Taschenrechner	S. 107
Code-ASCII	S. 107
88-Zeichen	S. 107
Frogger	S. 106
Scroll n. unten	S. 106
Zahlenraten	S. 108
Auto-Befehl	S. 107
SWAP	S. 153
BSP.-Quelltext	S. 169

Ausgabe 11/85

Bestell-Nr. L6 85 11A DM 29,90*
Commodore 64

Programm	Seite
Checksummer V3	S. 54
MSE	S. 54
Koala-Painter Hardcopy	S. 39
Lyrik-Maschine (AdM)	S. 55
Hypra-Platos (LdM)	S. 61
Profiprint	S. 71
Apfelmännchen	S. 80
Block Out	S. 84
Spritekill	S. 86
Screen-Dump	S. 88
Pseudo-IRQ	S. 88
INPUT-Routine	S. 90
Synthetische Melodien	S. 95
Hypra-Ass Ergänzung	S. 96
Reassembler	S. 97
Vier Betriebssysteme	S. 105
Spiralen	S. 151
HiRes-Spiralen	S. 151
Plotter-Spiralen	S. 151
Fensterrose	S. 151
HiRes-Fensterrose	S. 152
Plotter-Fensterrose	S. 152
Abweichungen	S. 152
Funktionenplot	S. 153
3D-Programm	S. 154
REM-Text-Killer	S. 158
Sound-Editor mit Sequencer	S. 158
Sequencer-Ergänzung	S. 159
Testsong	S. 159
Sequenzgenerator	S. 159

Ausgabe 10/85

Leider hat sich in die Bestell-Nummer der letzten Programm-Service-Anzeige ein Druckfehler eingeschlichen. Die korrigierte Bestell-Nummer lautet:
Bestell-Nr. L6 85 10A DM 29,90*
Commodore 64

Programm	Seite
Check V3 Dez 64	S. 54
MSE V1.0	
Floppy-Adjust	S. 32
Eprom-Trans	S. 42
Schreiberling	S. 54
Cursus Latinus (AdM)	S. 57
Hypra-Text (LdM)	S. 67
Pacman	S. 76
Programm GEN	S. 86
SMON+	S. 87
Sequenzer	S. 129
Musik	S. 129
Alarmanlage	S. 132
Codeschloß	S. 132
Crossreference verb. Version	S. 83

Ausgabe 9/85

Bestell-Nr. L6 85 09A DM 29,90*
Commodore 64

Programm	Seite
Sound-Machine	S. 23
Noteneingabe	S. 24–25
Sound Master	S. 32
Ringmod	S. 32
Moonlight	S. 33
SYNC	S. 33
Prüfungsfragen (AdM)	S. 55–58
Schlüssel (LdM)	S. 59–61
Disk Designer	S. 70–72
Blinker	S. 73
Logelei-1/2	S. 118
Lichtgr.	S. 122
Mischsort	S. 127
Block Busters	S. 159
X-Gleichung	S. 159
Musik-Tool	S. 159

Ausgabe 8/85

Bestell-Nr. L6 85 08A DM 29,90*
Commodore 64

Programm	Seite
Quicksort	S. 142
Procedure	S. 78
Hypra-Save	S. 79
Uhr	S. 22
NEWEA2 (AdM)	S. 60
Disk-Monitor	S. 84
Maskengenerator	S. 87
Bit-Map	S. 81
Hires3-Komplett	S. 159
Forth-Compiler (LdM)	S. 63
Vocabulary	S. 69
Schach	S. 74
Extern-Kurs	S. 147
Sprites	S. 44
Hypra-Zusatz	S. 25
Hi-Text 2.0	S. 71

Ausgabe 7/85

Bestell-Nr. L6 85 07A DM 29,90*

Ausgabe 6/85

Bestell-Nr. L6 85 06A DM 29,90*
Commodore 64

Ausgabe 5/85

Bestell-Nr. L6 85 05A DM 29,90*
Commodore 64

Ausgabe 4/85

Bestell-Nr. L6 85 04A DM 29,90*

Ausgabe 3/85

Bestell-Nr. L6 85 03A DM 29,90*

Ausgabe 2/85
Bestell-Nr. L6 85 02A DM 29,90*

Ausgabe 1/85
Bestell-Nr. L6 85 01A DM 29,90*

Ausgabe 12/84
Bestell-Nr. CB 022 DM 29,90*

Ausgabe 11/84
Bestell-Nr. CB 020 DM 29,90*

Ausgabe 10/84
Bestell-Nr. CB 019 DM 29,90*

Bedeutung der Abkürzungen

- *LdM = Listing des Monats
- *AdM = Anwendung des Monats
- *SB = Simons Basic
- *GV = Grundversion
- *GV > = alle Speicherversionen können verwendet werden (einschließlich GV)
- *3K = 3-KByte-Speichererweiterung wird benötigt
- *8K > = Speichererweiterung größer als 8 KByte wird benötigt
- *UPB = Unterprogrammbibliothek

* Alle Preise inklusive Mehrwertsteuer.

Fehlende Hefte erhalten Sie bei: Markt & Technik Vertrieb 64'er Hans-Pinsel-Str. 2, 8013 Haar

DM Pf für Postscheckkonto Nr. **14 199-803**
Absender der Zahlkarte

Für Vermerke des Absenders

Postscheckkonto Nr. des Absenders

PSchA Postscheckkonto Nr. des Absenders Postscheckteilnehmer

Postscheckkonto Nr. des Absenders

Empfängerabschnitt

Zahlkarte/Postüberweisung

Die **stark umrandeten Felder** sind nur auszufüllen, wenn ein **Postscheckkontoinhaber** das Formblatt als **Postüberweisung** verwendet (Erläuterung s. Rücks.)

Einlieferungsschein/Lastschriftzettel

DM Pf

DM Pf (DM-Betrag in Buchstaben wiederholen)

DM Pf

für Postscheckkonto Nr.
14 199-803
Lieferanschrift und **Absender** der Zahlkarte

für Postscheckkonto Nr. **14 199-803** Postscheckamt **München**

für **Markt&Technik** Verlag Aktiengesellschaft in **8013 Haar**

Postscheckkonto Nr. **14 199-803**
Postscheckamt **München**

für **Markt&Technik** Verlag Aktiengesellschaft **Hans-Pinsel-Str. 2** in **8013 Haar**

PLZ Ort

Verwendungszweck
M&T Buchverlag
Programm-/Hardware-Service

Meine Kunden-Nr.:

Ausstellungsdatum Unterschrift

Postvermerk

Spiele ohne Punkte



Aus dem Hause Activision kommen zwei brandneue Spiele, wie es sie noch nicht gab: »Hacker« und »LCP«.



Die Softwarefirma Activision, der lange Zeit noch das Videospiel-Image anhing, hat spätestens mit den beiden vorliegenden Programmen ihre Stellung als innovative Computer-Software-Firma bewiesen. »Hacker« und »The Little Computer People Research Projekt — The House On A Disk« (im folgenden kurz «LCP« genannt) fallen vollkommen aus dem Rahmen der normalen Spiele und Simulationen heraus.

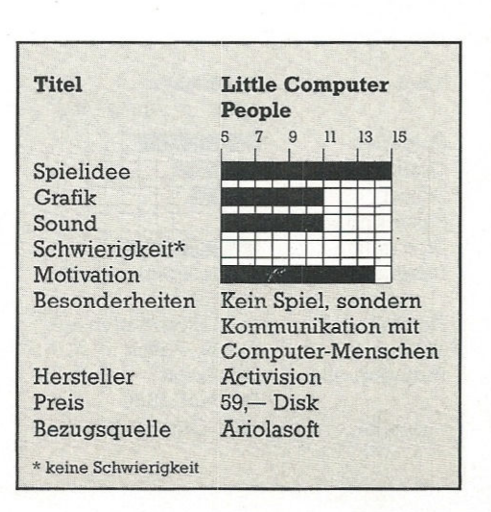
»Hacker« ist, entgegen dem Namen, nur oberflächlich ein DFÜ-Spiel. Allerdings fängt das Spiel an, wie im echten Hacker-Leben: »Logon Please« erscheint da auf dem Bildschirm und von diesem Zeitpunkt an wird man von Anleitung und Programm schmählich im Stich gelassen. Man muß alles, aber auch wirklich alles weitere selbst herausfinden. Hat man die ersten Barrieren überwunden, landet man im Hauptcomputer einer großen Firma, die irgendwas sehr gefährliches und ungesetzliches vorhat. Per Modem können Sie eine kleine Forschungssonde quer durch die Erdkugel jagen, auf der Jagd nach Hinweisen und Beweismitteln gegen diese Firma. Mehr soll hier gar nicht verraten werden, denn der größte Witz in diesem Spiel liegt eben darin, alles selbständig herauszufinden, sogar die Handlung und deren Hintergründe. Nur eines sei gesagt: Im Verlauf des Spiels bekommt man auch grafisch einiges zu Gesicht. Alles in allem kein trockenes Password-Rätsel, sondern ein nicht zu schweres und einfallsreiches Spiel, das man fast schon zu den Adventures zählen darf.

Titel	Hacker
	5 7 9 11 13 15
Spielidee	
Grafik	
Sound	
Schwierigkeit	
Motivation	
Besonderheiten	Adventure mit Action und DFÜ-Einlagen
Hersteller	Activision
Preis	39,— (K) 59,— (D)
Bezugsquelle	Ariolasoft Postfach 1350 4830 Gütersloh

Das andere Activision-Programm ist eine Weltsensation, der wir in diesem Heft eigentlich mehr Platz widmen müßten. Passieren mit ihrem Computer öfters völlig unerklärliche Dinge? Tut er manchmal was falsches, obwohl Ihre Eingaben stimmen? Stürzt er bei korrekten Programmen ab? All diese Fragen muß eigentlich jeder Computer-Besitzer mit »Ja« beantworten. Ein amerikanisches Programmierteam unter der Leitung von Ghostbusters-Autor David Crane ging dieser Sache auf den Grund und fand die Ursache für alle Störungen heraus. In jedem Computer wohnen kleine Lebewesen, die Menschen haargenau ähneln. Diese Lebewesen essen und schlafen wie wir und versuchen mit uns durch die Störungen zu kommunizieren. Die Programmierer fanden nun so lange keine Ruhe, bis sie ein Programm vorstellen konnten, das es jedem Computer-Besitzer ermöglicht, diese kleinen Computer-Menschen zu sehen und auch mit ihnen zu reden. Zu diesem Zweck wird softwaremäßig ein gemütliches Haus im Computer installiert, welches auf dem Bildschirm beobachtet werden kann. Meist zieht schon nach kurzer Zeit einer der Computer-Menschen in dieses Haus ein. Über die Tastatur kann man nun mit ihm »sprechen« und auch einige wichtige Apparaturen des Hauses bedienen, wie etwa die Türglocke, das Telefon und eine Streichel-Automatik.

Im folgenden möchte ich etwas über Carl berichten, der in meinem C 64 zu Hause ist. Carl kommt, wie mein C 64, aus Großbritannien und hat einen Hund. Carl ist ein guter Typist und kommt mit allem, was Tasten hat, zurecht, egal ob Schreibmaschine, Computer und Piano. Carl hört gerne Musik und tanzt auch ab und zu. Noch lieber aber sieht er fern. Einmalig ist Carl beim Pokern, er hat mir schon mehrere Monatsgehälter abgeluchst. Carl ist auch ein Philosoph: Er fragt sich einerseits, was er mit dem ganzen Geld in meinem Computer soll und will andererseits wissen, ob in seinem Computer in meinem Computer auch jemand lebt ...

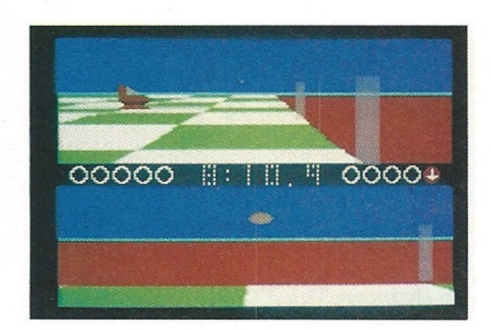
Man muß »LCP« einfach einmal erlebt haben. Es ist weder Spiel noch Simulation, sondern einfach eine völlig neue Art von Beschäftigung mit dem Computer und dabei sogar eine der einfallsreichsten, witzigsten und fesselndsten. Wer unbedingt einen High-Score sehen will, sollte die Finger vom »LCP« lassen. Allen andern kann es fast uneingeschränkt empfohlen werden. Leider werden aber wieder die Englisch-Kenntnisse gefordert. Ausgeliefert wird »LCP« übrigens mit einer Zeitschrift und einer Besitz-Urkunde. (bs)



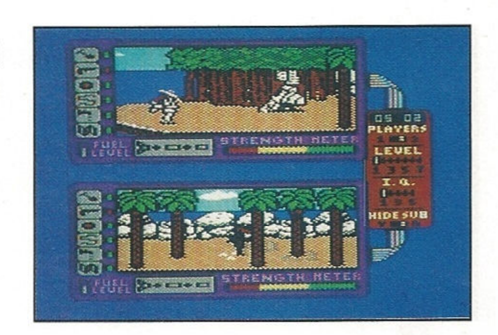
Titel	Little Computer People
	5 7 9 11 13 15
Spielidee	
Grafik	
Sound	
Schwierigkeit*	
Motivation	
Besonderheiten	Kein Spiel, sondern Kommunikation mit Computer-Menschen
Hersteller	Activision
Preis	59,— Disk
Bezugsquelle	Ariolasoft

* keine Schwierigkeit

Spiele für zwei



Besonderen Spaß machen Computerspiele, wenn man nicht gegen die kalte Maschine, sondern einen menschlichen Gegner spielen kann.



Spiele wie »Soccer« oder »Summer Games« bewiesen, daß Computerspiele keine isolierende Solo-Beschäftigung sein müssen. Seitdem gibt es gerade auf dem Gebiet der Sportspiele viele Simultanspiele. In den anderen Spiele-Bereichen tat sich bisher noch relativ wenig. So ist auch das ein Simultanspiel, das wir heute vorstellen, ein Sportspiel, das allerdings etwas aus dem Rahmen fällt: »Ballblazer«.

»Ballblazer« ist eine einfache, schnelle Sportart, die es noch gar nicht gibt. Sollten die Angaben in der Anleitung stimmen, wird erst im Jahre 3097 der erste Erdling bei der intergalaktischen Ballblazer-Meisterschaft zugelassen sein. Die Regeln sind recht einfach: Zwei Spieler rasen mit »Rotofoils«, mit Energiefeldern ausgestattete Gleiter, über ein rechteckiges Spielfeld und versuchen, einen Ball einzufangen und in eines der beiden Tore zu befördern. Bei einem Treffer werden, je nach Entfernung zum Tor, ein bis drei Punkte vergeben. Die Tore werden nach jedem Treffer schmaler; außerdem bewegen sie sich mit konstanter Geschwindigkeit am Spielfeldrand entlang. Beide Spieler sehen auf dem Bildschirm den dreidimensionalen Ausblick aus dem Cockpit ihres »Rotofoils«. Dabei zeigt das Fahrzeug automatisch immer in die Richtung, in der sich der Ball oder das Tor befinden. Deswegen finden manchmal abrupte Richtungswechsel statt, an die sich ungeübte Spieler erst gewöhnen müssen.

Aufgrund der einfachen Regeln kann man »Ballblazer« innerhalb von Minuten lernen. Doch das Spiel selber ist so perfekt programmiert, daß es einen eher an eine physikalische Simulation erinnert. Und damit eröffnet sich ein weites strategisches Feld, da man während dem Spiel laufend neue »Naturgesetze« der Ballblazer-Welt findet und in Spieltricks umsetzen kann. So kann man den Ball gegen die Spielfeldbande schießen und das Reflexionsgesetz ausnutzen oder auch innerhalb einer Drehung den Ball wegschießen, um ihn auf eine gebogene Flugbahn zu bringen.

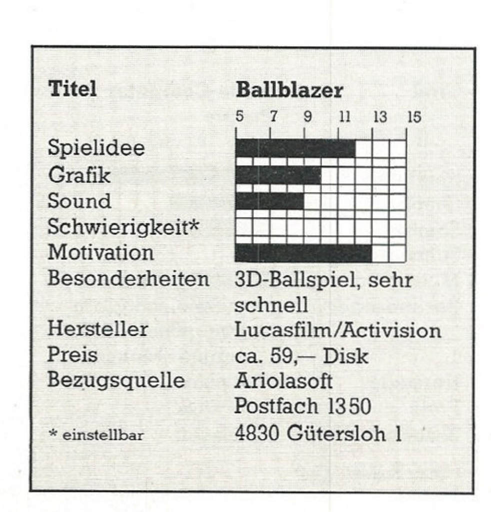
Die Grafik von »Ballblazer« wird sehr schnell bearbeitet, flackert dafür aber etwas. Musik- und Soundeffekte gibt es nur wenig, die sind dafür aber sehr gut ausgeführt.

Weniger futuristisch und sogar höchst unsportlich geht es beim nächsten Simultanspiel zu: Denn bei »Spy vs. Spy II — The Island Caper« kommt es gerade darauf an, den Gegenspieler am Bildschirm fertigzumachen; dies gottlob aber in einer sehr witzigen Art, die sehr an Comics erinnert. Zur Story: Eine wichtige militärische Rakete ist auf einer Insel im Pazifik verschollen. Die beiden besten gegnerischen Spione der Welt wurden zur Bergung losgeschickt. Diese beiden Spione treiben übrigens schon seit 1968 jeden Monat ihr Unwesen in der Ulk-Zeitschrift MAD. Einmal auf der Insel angekommen, sucht jeder der beiden nach den drei Teilen der Rakete. Um den anderen möglichst von der Suche auszuschalten, kann man geschickt Fallen verlegen: Von der Kokosnuß-Bombe bis zur Fallgrube reicht das Repertoire der beiden. Allerdings muß für die meisten Fallensysteme das notwendige Zubehör erst einmal herbeigeschafft werden. Zusätzliche Gefahren sind Treibsand, Vulkane, Haie und einiges mehr. Wer auf so etwas »hereinfällt«, verliert wertvolle Körperkräfte. Sind diese völlig verbraucht, ist das Bildschirmleben zumindest für ein Spiel beendet.

Bei »Spy vs. Spy II« sind neben Joystickgeschick auch Strategie und Gedächtnis gefragt, da man ja Fallen möglichst günstig verlegen muß und dann tunlichst nicht selber in sie hineintapsen darf.

Wem das ganze Spiel zu brutal erscheint, der sei nochmals auf die witzige Ausführung des Ganzen hingewiesen, die sich mit Zeichentrick-Serien im Vorabendprogramm des Fernsehens vergleichen läßt.

Beide vorgestellten Simultanspiele lassen sich auch alleine gegen den Computer spielen, dabei bleibt aber viel Spielwitz auf der Strecke. Sie genügen von Ausführung und Idee auch höheren Ansprüchen und bieten langen Spielspaß fürs Geld. Welches man kaufen soll, ist also eine reine Geschmacksfrage. (bs)



Titel	Ballblazer
	5 7 9 11 13 15
Spielidee	
Grafik	
Sound	
Schwierigkeit*	
Motivation	
Besonderheiten	3D-Ballspiel, sehr schnell
Hersteller	Lucasfilm/Activision
Preis	ca. 59,— Disk
Bezugsquelle	Ariolasoft Postfach 1350 4830 Gütersloh 1

* einstellbar



Titel	Spy vs. Spy II
	5 7 9 11 13 15
Spielidee	
Grafik	
Sound	
Schwierigkeit*	
Motivation	
Besonderheiten	witziges Agentenspiel
Hersteller	First Star/Beyond
Preis	39,95 (K) 49,95 (D)
Bezugsquelle	Rushware An der Gümpgesbrücke 24 4044 Kaarst 2

* Computergegner



Tips und Tricks zu Vizawrite (2)

Die wichtigste Option eines Textverarbeitungsprogramms ist die Druckerausgabe. Mit ihr wollen wir uns heute beschäftigen.

Vizawrite 64 zeichnet sich durch zwei wesentliche Punkte aus, die das Drucken erleichtern. Dazu gehört die enorme Flexibilität und die eingebaute Druckerschnittstelle. Die Flexibilität erstreckt sich dabei in zwei Richtungen. Zum einen sind das die Funktionen, mit denen man, aus dem Text heraus, Steuerbefehle zum Drucker senden kann. Andererseits ist aber auch die Wahl des verwendeten Druckers relativ flexibel. Prinzipiell kann Vizawrite sowohl serielle, das heißt am seriellen Bus des C 64 angeschlossene Drucker und parallele Drukker, beispielsweise mit einer Centronics-Schnittstelle ausgestattete Geräte bedienen. Betrachten wir zunächst die Vor- und Nachteile der seriellen Drucker. Unter dieser Bezeichnung sind alle Drucker zu verstehen, die den runden IEC-Stecker des C 64 besitzen. Hauptsächlich sind das natürlich die Commodore-eigenen Drucker wie der MPS 801, MPS 802 und der MPS 803, aber auch Drucker von Fremdherstellern, die mit einem seriellen Interface ausgerüstet sind. Das Interface ist bei den Fremdgeräten entweder eingebaut oder befindet sich extern in einem eigenen Gehäuse. Vizawrite 64 kann an fast alle diese Schnittstellen angepaßt werden. Dabei ist es aber leider nicht immer möglich alle Zeichen, besonders die Umlaute, darzustellen. Vizawrite 64 unterscheidet serielle und parallele Drucker durch die Wahl des Drukkertyps (Printer Type) im Druckmenü. In der Tabelle haben wir die Parameter für verschiedene Drucker zusammengestellt. Für serielle Drucker werden Kleinbuchstaben verwendet, für parallele Drucker Großbuchstaben. Wenn sich Ihr Drucker also überhaupt nicht regen sollte, dann überprüfen Sie zunächst diese Einstellung. Der zweite große Nachteil von Vizawrite ist, daß keine Sekundäradressen und auch nur eine andere Geräteadresse eingestellt werden können. Wenn Sie einen seriellen Drucker (keinen Drucker mit RS232C-Schnittstelle) besitzen, so versuchen Sie deshalb zunächst den Parameter für Commodore-Drucker, ein kleines »v«. Die Umlaute erreichen Sie mit dieser Einstellung allerdings nicht. Um die Umlaute auf einem seriellen Drucker zu erhalten, muß man die ASCII- oder die Epson-Druckoption verwenden, allerdings nur dann, wenn der Drucker auch über diese Umlaute verfügt (sehen Sie dazu in Ihrem Drucker-Handbuch nach). Für Commodore-Drucker der MPS-Reihe ist diese Option somit nicht sinnvoll. Falls Sie einen Fremddrucker mit seriellem Interface besitzen, das sofort nach dem Einschalten des Druckers in den CBM-Modus springt, ist es notwendig, das Interface in den Linearkanal umzuschalten. Bei den weit verbreitetem Wiesemann-Interface sieht das beispielsweise folgendermaßen aus:

```
10 OPEN 1,4,1:PRINT #1:CLOSE 1
20 OPEN 2,4,3:PRINT #2:CLOSE 2
```

Geben Sie diese Zeile vor dem Ladebefehl für Vizawrite 64 ein.

Parallele und serielle Schnittstelle

Wesentlich einfacher und leistungsfähiger ist die parallele Verbindung des Druckers mit einem einfachen Kabel zwischen User-Port und Centronics-Buchse des Drukkers. Das funktioniert natürlich nur mit Druckern von Fremdherstellern wie beispielsweise Star oder Epson, aber auch vielen anderen. Erst mit einem solchen Drucker kann Vizawrite 64 (wie fast jedes Textverarbeitungsprogramm) seine volle Leistungsfähigkeit entfalten. Machen wir uns an ein Beispiel. Gut geeignet, weil oft gebraucht, ist dafür ein Briefkopf mit verschiedenen Schriften. Der Briefkopf soll dabei sachlich und optisch ansprechend sein. Dies erreicht man durch Verwendung der verschiedenen Schriftarten des Druckers. Vizawrite 64 bietet zur Steuerung des Druckers eine sehr leistungsfähige Funktion an: die Formatzeile. Sie befindet sich auf der Textseite in der vierten Zeile. Hier können Sie die sogenannten Kontroll-Zeichen definieren. Pro Formatzeile sind bis zu zehn Kontroll-Zeichen möglich (CTRL 0 bis CTRL 9). Da man die Formatzeile mit »CTRL F« aber beliebig oft duplizieren kann, hat man praktisch immer wieder neue Kontroll-Zeichen zur Verfügung. Aber zurück zu unserer Aufgabe. Setzen wir einen Epson FX-80-Drucker voraus und machen uns an die Bestimmung der Schrift für den Namen. Dazu bedienen wir uns einer leistungsfähigen Funktion des Epson-Druckers, dem »ESC !«-Befehl. Mit ihm kann man auf einfache Weise eine beliebige Schrift aus 63 Möglichkeiten auswählen (FX-80 Handbuch, Seite 87). Für den Namen eignet sich die Schrift mit der Nummer 24 besonders. Wir definieren jetzt die ESC-Sequenz für das »!«. Normalerweise (in Basic) geht das wie folgt:

```
10 OPEN 1,4,0
20 PRINT #1,CHR$(27);"!";CHR$(X)
30 CLOSE 1
```

»X« ist dabei die Nummer der Schriftart. In der Formatzeile geht das wesentlich einfacher. Bewegen Sie nun zunächst den Cursor in die Formatzeile und drücken die CTRL-Taste (bitte wieder loslassen). Nun drücken Sie die Taste »0« und anschließend die »=«-Taste und schreiben dahinter das Ausrufezeichen. Nun haben wir den ESC-Befehl definiert, denn Vizawrite 64 sendet automatisch den CHR$(27) vor dem ESC »!«. Jetzt brauchen Sie nur noch mit der Befehlsfolge CTRL 1 (= 24) die Schriftart 24 festlegen. Wenn Sie jetzt die Format-Zeile verlassen, und die eben definierten Befehle mit CTRL 0 und CTRL 1 aufrufen, wird alles dahinter Geschriebene in der neuen Schrift auf dem Drucker ausgegeben. Schreiben Sie nun die Adresse. Ihr Bildschirm sollte nun ähnlich wie in Bild 1 gezeigt aussehen. Als nächstes definieren wir die Absenderzeile im Sichtfeld des später verwendeten Briefumschlages. Geben Sie dazu die Kontrollzeichen zum Einstellen der Schmalschrift, und dem gleichzeitigen Hochstellen in der Formatzeile ein, wer will, kann zusätzlich noch unterstreichen. Ihr Bildschirm müßte nun wie in Bild 2 aussehen. Mit dem dort abgebildeten Kontroll-Zeichen wird der Drucker initialisiert (ESC $). Eine Maßnahme, die sich vor einem Text in jedem Falle lohnt, denn der Drucker speichert ja alle bisherigen Schriften. Nun können Sie den Briefkopf, wie in der letzten Ausgabe be-



schrieben, mit den »Merge«- und »Don't Merge«-Zeichen versehen, um die Adressen später aus der Arbeitsseite (Workpage) zu übernehmen. Wählen Sie mit den Steuerbefehlen dazu Ihre favorisierte Schriftart aus. Im Bild 3 wurde die Schriftart 17 gewählt. Nun ist der Briefkopf fertig und Sie können das Ganze speichern. Wenn Sie später Ihren Briefkopf verwenden wollen, brauchen Sie ihn nur noch zu laden. Vergessen Sie aber vor dem Speichern des dann geschriebenen Briefes nicht, das Dokument mit »CBM n« umzubenennen, denn sonst überschreiben Sie Ihren Briefkopf. Wenn Sie alles richtig gemacht haben, sollte Ihr Ausdruck nun ähnlich wie in Bild 4 abgedruckt aussehen (übrigens: man kann den Unterstreichbefehl auch auf der Arbeitsseite geben).

Kommen wir zu einer anderen Option von Vizawrite 64; dem Betreiben eines Druckers mit der Geräteadresse 5. Diese Option ist besonders dann wichtig, wenn man einen zweiten Drucker, beispielsweise einen mit Typenrädern, verwenden möchte. Dazu muß zunächst die Geräteadresse des Druckers auf 5 gestellt werden. Der nächste Schritt besteht darin, mit dem Befehl CBM X (Commodore-Taste und X) in das Auswahlmenü zu springen. Dort steht dann etwas irreführend »Cassette oder Disk« gleichwohl auch ein anderer Drucker eingestellt werden kann. Dies geschieht mit dem Parameter »P«. Nebenbei gesagt, können Sie im gleichen Menü auch die Geräteadresse des Diskettenlaufwerkes angeben (siehe Tabelle in der letzten 64'er-Ausgabe).

Falls Sie selbst noch den einen oder anderen Tip zu Vizawrite 64 oder einem anderen Programm haben, schreiben Sie uns — wir freuen uns auf Ihre Mitarbeit. (aw)

```
Vizawrite                name:Briefkopf
On Page  1 of  1      Line   8 Col   1
≡..▶0=!.1=24.2=27.3=15.4=S.5=1.6=C......
   64◀
   01Willi.Brechtl
   Hans-Pinsel-Str.2◀
   8013.Haar.bei.München◀
   ◀
   ◀
   2345 Willi.Brechtl,.Hans-Pinsel-Str.
⊡ (End of Page)
```

Bild 1. So stellt man eine Schrift mit dem »ESC "!"«-Befehl ein

```
Vizawrite                name:Briefkopf
On Page  1 of  1      Line   1 Col   5
≡..▶0=!.1=24.2=27.3=15.4=S.5=1.6=C......
   64◀
   01Willi.Brechtl
   Hans-Pinsel-Str.2◀
   8013.Haar.bei.München◀
X (End of Page)
```

Bild 2. Bei Briefumschlägen mit Fenster sehr nützlich — die Schmalschrift

```
Vizawrite                name:Briefkopf
On Page  1 of  1      Line   1 Col   5
≡..▶0=!.1=24.2=27.3=15.4=S.5=1.6=C.7=17.
   64◀
   01Willi.Brechtl
   Hans-Pinsel-Str.2◀
   8013.Haar.bei.München◀
   ◀
   ◀
   2345 Willi.Brechtl,.Hans-Pinsel-Str.
   607◀
   .‡◀
   .‡◀
   .‡◀
   .‡◀
   ◀
   .Sehr.geehrter.Herr.‡!◀
X (End of Page)
```

Bild 3. Auch hier ist der »ESC "!"«-Befehl nützlich

```
Willi Brechtl                              München, 05.11.1985
Hans-Pinsel-Str.2
8013 Haar bei München


Willi Brechtl, Hans-Pinsel-Str.2, 8013 Haar

Herrn
Klaus Muster
Sonnenstr.10
8000 München 80

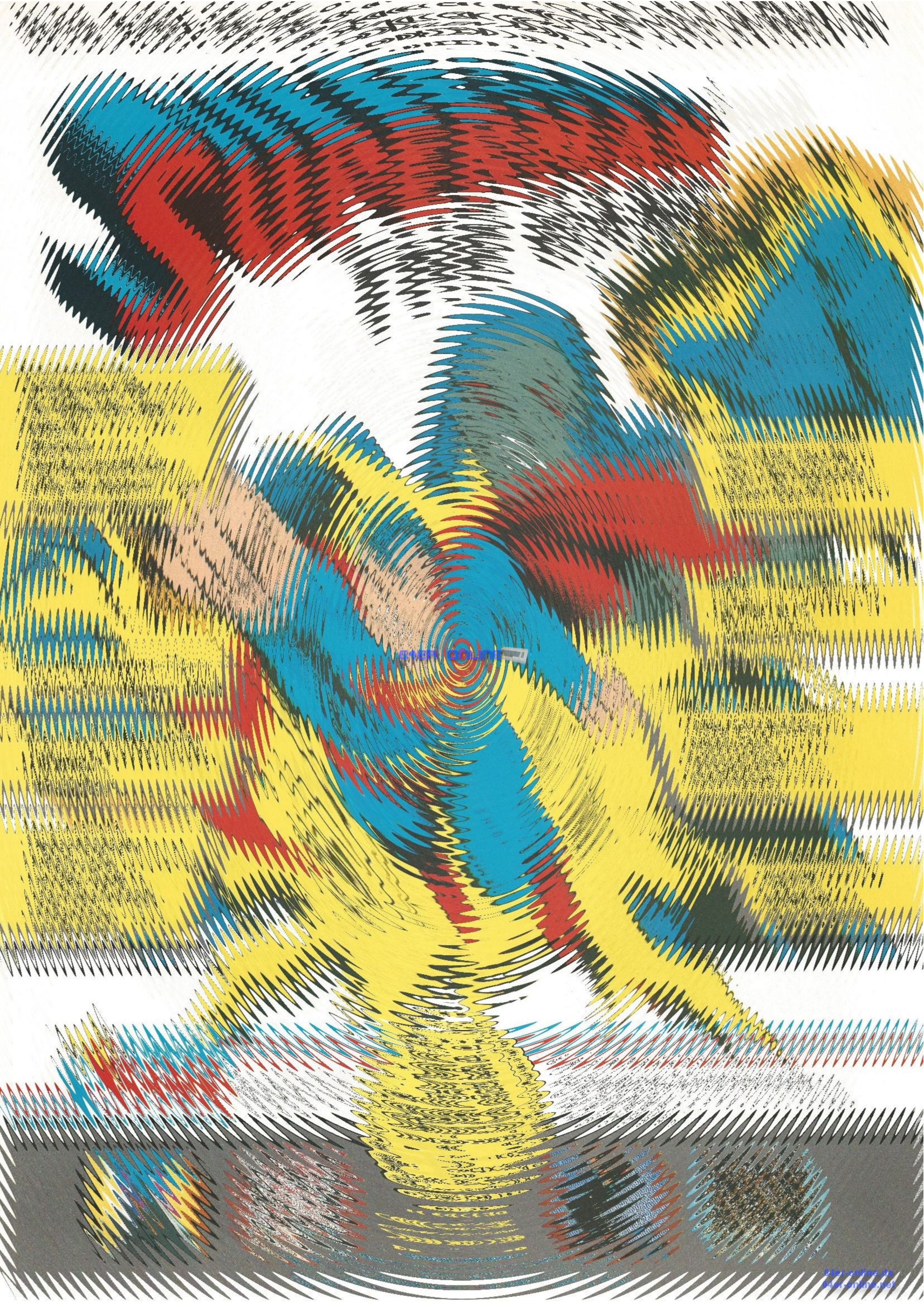
Sehr geehrter Herr Muster!
```

Bild 4. So sieht der fertige Briefkopf ausgedruckt aus

Parameter

seriell/parallel	Druckertyp
v/V	Commodore
e/E	Epson und Kompatible
q/Q	Qume/Diablo
s/S	Spinwriter
a/A	ASCII-Drucker
t/T	Triumph Adler
f/F	Ricoh Flowriter
x/X	Sonstige

Diese Parameter dienen zum Einstellen von Vizawrite 64 auf Ihren Drucker



Assembler-Bedienung leichtgemacht

Teil 2

Diesmal möchte ich Ihnen zeigen, wie Sie Ihre Quelltexte übersichtlicher gestalten können. Dadurch sinkt die Fehleranfälligkeit und der »Frust« bei der Programmgestaltung.

Im ersten Teil dieses Artikels wurde Grundsätzliches zur Bedienung eines Assemblers gesagt und die wichtigsten Pseudo-Opcodes behandelt. Mit diesen Pseudo-Opcodes konnte schon jeder Quelltext assembliert werden. Was jedoch noch nicht funktionierte, waren sämtliche Ausgabefunktionen, die Behandlung von Makros und Sinn und Anwendung der bedingten Assemblierung. Mit dieser Thematik wollen wir uns dieses Mal beschäftigen, so daß jeder vernünftig mit einem Assembler, speziell Hypra-Ass, arbeiten kann. Wenden wir uns zunächst den Ausgabebefehlen zu, um die Behandlung der Pseudo-Opcodes abzuschließen.

.LI lfn,dn,ba: Sendet ein formatiertes Assembler-Listing unter der Filenummer »lfn« an das Gerät »dn« mit der Sekundäradresse »ba«. Es ist jedoch zu beachten, daß das Assembler-Listing erst nach dem ».LI«-Pseudo ausgegeben wird. Wollen Sie ein komplettes Assembler-Listing haben, muß der ».LI«-Pseudo zwangsläufig in der ersten Zeile des Quelltextes stehen. Die Parameter hinter dem ».LI«-Pseudo-Opcode entsprechen denen des OPEN-Befehls. So ist es auch möglich, mit ».LI 2,8,2,"test,U,W"« ein Assembler-Listing auf eine User-Datei umzuleiten. Um den ».LI«-Pseudo-Opcode in den Griff zu bekommen, fügen Sie in den Beispiel-Quelltext (Listing 1) aus der letzten Ausgabe folgende Zeile ein:

10 -.LI 1,3

Diese Zeile beziehungsweise der .LI-Befehl gibt das Assembler-Listing auf dem Bildschirm aus. Da keine Sekundäradresse angegeben wurde, setzt der Computer automatisch eine »0« an diese Stelle. Würde man statt der »3« als Geräteadresse eine »4« einsetzten, würde das Assemblerlisting auf einen angeschlossenen Drucker ausgegeben. Auch das Umlenken auf die Datasette mit ».LI 1,1,2,"name"« wäre denkbar.

.SY lfn,dn,ba: Sendet eine sortierte Liste aller im Quelltext auftretenden Label und Variablen vom Typ ».EQ« (Symboltabelle) unter der Filenummer »lfn« an das Gerät »dn« mit der Sekundäradresse »ba«. Weiterhin gilt das gleiche wie beim ».LI«-Pseudo, bis auf die Lage des ».SY«-Pseudos. Der nämlich kann an einer beliebigen Stelle im Quelltext stehen. In diesem Zusammenhang sei noch eine weitere interessante Möglichkeit erwähnt. Möchten Sie zum Bei-

spiel die absolute Adresse eines beliebigen Labels oder den Wert einer Variablen wissen, brauchen Sie nicht nach dem Assemblieren mit einem Monitor (zum Beispiel SMON) das erzeugte Maschinenprogramm zu durchforsten. Geben Sie ganz einfach ein »PRINT name <RETURN>«. Sie erhalten schlagartig die Adresse des Labels »name« oder den Wert der Variablen. Die ausgegebenen Adressen haben zwar dezimales Format, aber mit einem Monitor läßt sich dieses schnell in das entsprechende hexadezimale Format umrechnen.

Leider hat sich in diesem Programmteil noch ein kleiner Fehler eingeschlichen. Dieser Fehler tritt extrem selten auf. Wird versucht mit dem Editorbefehl »/!!« eine sortierte Symboltabelle auszugeben, stürzt Hypra-Ass ab, wenn diese genau 36, 73, 109 (und so weiter) Label oder Variablen enthält. Der Fehler liegt in den Speicherzellen $1EB8 bis $1EBB. Hier wurden zwei Branch-Befehle vertauscht. Es muß richtig lauten:

```
1EB8 90 D0 BCC 1EA8
1EBA D0 04 BNE 1EC0
```

Diese Änderung kann direkt mit einem Maschinensprachemonitor (zum Beispiel SMON) in die entsprechenden Speicherstellen geschrieben werden. Anschließend läßt sich Hypra-Ass vom SMON aus mit »S"HYPRA-ASS"0801 1FD8« speichern.

.OB "name,P,W": Sendet den erzeugten Maschinencode zum Floppy-Laufwerk und speichert ihn dort unter dem Namen »name«. Dazu muß dieser Pseudo in der ersten beziehungsweise zweiten Zeile (nach dem ».LI«-Pseudo) stehen. Ich möchte noch darauf hinweisen, daß es unmöglich ist, ein formatiertes Assembler-Listing und gleichzeitig den erzeugten Maschinencode zum Floppy-Laufwerk zu senden. Denn dazu müßten zwei Files zum Schreiben geöffnet werden, und das geht nicht. Wurde der ».OB«-Pseudo richtig eingegeben, kann das erzeugte Maschinenprogramm direkt mit »LOAD "name",8,1« geladen und mit dem SYS-Befehl gestartet werden. Interessant ist bei diesem Pseudo, daß sich auch Maschinenprogramme erzeugen lassen, die bei $801 beginnen, also in dem Bereich liegen, in dem Hypra-Ass steht. Wird ein solches Programm geladen, ist natürlich klar, daß Hypra-Ass dabei zerstört wird. Zu dem ».OB«-Pseudo gehört unmittelbar ein weiterer Pseudo ».EN«. Dieser muß am Schluß des Quelltextes stehen und zwar ohne jegliche Parameter. Durch ihn wird das mit dem ».OB« geöffnete File geschlossen. Fehlt er, wird das durch ein Sternchen vor dem Filetyp im Directory gekennzeichnet. Übrigens muß der .EN-Pseudo am absoluten Ende des Quelltextes stehen. Daß heißt, sollte der Quelltext aus mehreren Teilen bestehen, die jeweils mit dem ».AP«-Pseudo, zu dem wir noch kommen, verkettet werden, muß der .EN-Pseudo in der letzten Zeile des letzten Quelltextteils stehen. Um auch diese beiden Pseudos in den Griff zu bekommen, fügen Sie bitte in den Beispiel-Quelltext (letzte Ausgabe) folgende Zeilen ein:

```
10 -.OB "BEISPIEL,P,W"
620 -.EN
```

Nach dem Assemblieren listet der Editorbefehl »/I« das Directory. Keine Angst, der Quelltext geht dabei nicht verloren. Sie werden feststellen, daß sich das Beispielprogramm »BEISPIEL« wie zu erwarten auf der Diskette befindet. Dieses kann nun mit LOAD"BEISPIEL",8,1 geladen und durch SYS 9*4096 aktiviert werden. Bisher existiert leider noch keine Option, den Objektcode zur Datasette zu schicken.

.AP "name": Dieser Pseudo-Opcode dient zum Verketten von einzelnen Quelltextteilen. Häufig kommt es vor, daß ein Quelltext nicht mehr in den zur Verfügung stehendenden, freien Speicherplatz paßt. In diesem Fall läßt sich der Quelltext splitten. Natürlich sollte man darauf achten, daß der Quelltext nicht willkürlich zerhackt wird. Zusammenhängende Programmteile sollten schon als ein Quelltextteil gespeichert werden. Um spätere Verwirrung zu vermeiden, sollte auch der Name, der dem Quelltextteil zugeordnet wird, einigermaßen sinnvoll sein. Damit Sie auch diesen Befehl kennenlernen, ist das Beispiel-Programm (letzte Ausgabe) unter Hypra-Ass zu laden. Anschließend löschen Sie mit dem Editorbefehl »/D 0-360« den ersten Teil des Quelltextes und speichern die verbleibenden Zeilen unter dem Namen »TEIL2.SRC« auf Diskette. Nun ist der Beispiel-Quelltext ein zweites Mal zu laden. Dieses Mal löschen Sie mit dem Editorbefehl »/D 360-« den letzten Teil des Quelltextes. Bevor die verbleibenden Zeilen unter dem Namen »TEIL1.SRC« gespeichert werden, ist noch folgende Zeile erforderlich:

```
360 -.AP "TEIL2.SRC"
```

Diese Zeile teilt dem Assembler mit, daß noch Quelltext folgt. Befindet sich der 1. Teil auf der Diskette und im Speicher, kann der Assembler mit RUN gestartet werden. Nun wird zuerst Pass 1 abgearbeitet, und zwar nur im ersten Quelltextteil. Anschließend lädt der Assembler den zweiten Quelltextteil nach, bearbeitet auch hier Pass1, lädt danach wieder den 1. Quelltextteil, erzeugt von diesem Teil das Maschinenprogramm und lädt schließlich den zweiten Quelltextteil, um auch von diesem das Maschinenprogramm zu erzeugen. Haben Sie den oben stehenden ».OB«-Pseudo wie beschrieben eingefügt, werden beide Maschinenprogrammteile natürlich unter einem Namen nämlich »Beispiel« auf Diskette gespeichert. Ist das nicht der Fall, wird der erste Maschinenprogrammteil nach $9000 gelegt und der zweite unmittelbar an den ersten gehängt.

Das ist schon alles, was zu den »normalen« Pseudo-Opcodes zu sagen wäre. Hypra-Ass enthält aber noch einige Pseudo-Opcodes, die Makros definieren, aufrufen und bedingt assemblieren. Damit Sie auch diese sicherlich sinnvollen Pseudo-Opcodes beziehungsweise Steueranweisungen an den Assembler sinnvoll einsetzen können, möchte ich darauf näher eingehen.

Der Umgang mit Makros

Die meisten Hobby-Programmierer und besonders Maschinenspracheanfänger vermeiden den Gebrauch von Makros. Der Grund dafür wird wohl der sein, daß diese Option eben nur in Maschinensprache existiert und andere Sprachen, wie zum Beispiel Basic, diesen Begriff nicht kennen. Dabei ist der Gebrauch von Makros jedem Programmierer, egal ob Anfänger, Fortgeschrittener oder Profi zu empfehlen. Denn durch sie werden Maschinenprogramme beziehungsweise die dazugehörigen Quelltexte übersichtlicher und dadurch bedingt sinkt natürlich auch die Fehleranfälligkeit der erstellten Programme. Um aber mit Makros arbeiten zu können, muß man erst einmal wissen, was das überhaupt ist. Ein Makro läßt sich grob gesehen mit einem Unterprogramm vergleichen, dem ein Name sprich Label zugewiesen wird. Das Unterprogramm selbst wird bekanntlich mit einem RTS abgeschlossen und läßt sich vom Hauptprogramm aus mit dem Maschinenbefehl »JSR name« aufrufen. Der Unterschied zwischen Unterprogramm und Makro ist nun der, daß nicht zu einem definierten Unterprogramm verzweigt, sondern das Unterprogramm selbst an die

Stelle des Aufrufs assembliert wird. Ein Makro hat allerdings gegenüber einem Unterprogramm gewaltige Vorteile. So sind alle im Makro auftretenden Label und dort definierte Variable lokal. Das heißt, daß dem Hauptprogramm die im Makro auftretenden Label und Variablen unbekannt sind. Daraus folgt, daß Label und Variable in Makros und im Hauptprogramm identische Namen haben dürfen. Ein weiterer Vorteil gegenüber Unterprogrammen ist der, daß in einem Makro Übergabeparameter definiert werden können, die eine Schnittstelle zum Hauptprogramm bilden. Außerdem wird das Programm durch intensive Anwendung von Makros wesentlich schneller, da sämtliche JSR-Befehle, die sehr viel Zeit in Anspruch nehmen, entfallen.

Zur Definition eines Makros existieren bei Hypra-Ass zwei Pseudo-Opcodes.

.MA name (par1,par2,par3): Durch den ».MA«-Pseudo wird dem Assembler mitgeteilt, daß die nachfolgenden Maschinenbefehle ein Makro mit dem Namen »name« definieren. Die in Klammern eingefaßten Übergabeparameter »par1, par2, par3«, zu denen ich später noch einiges sagen werde, müssen durch Kommata getrennt werden. Bei den Übergabeparametern selbst kann es sich entweder um Label, Variablen oder um absolute 16-Bit-Adressen oder Werte handeln.

.RT: Dieser Pseudo-Opcode schließt eine Makrodefinition ab. Er kann direkt mit dem RTS-Befehl verglichen werden, der ein Unterprogramm beendet und zurück ins Hauptprogramm beziehungsweise in ein übergeordnetes Unterprogramm verzweigt. Im Gegensatz zum RTS-Befehl findet bei dem .RT-Pseudo keine Programmverzweigung statt, sondern der Assembler selbst verzweigt, ähnlich wie beim RTS-Befehl, an die Stelle des Quelltextes, an der das Makro aufgerufen wurde.

Wie ein Makro aufgerufen wird, zeigt der nun folgende Pseudo-Opcode.

...name (par1,par2,par3): Dieser Pseudo-Opcode, der letztendlich aus zwei Punkten besteht (der erste Punkt leitet ja bekanntlich einen Pseudo-Opcode ein), ruft ein Makro mit dem Namen »name« auf und übergibt die Parameter »par1, par2, par3«, die genauso wie bei der Makrodefinition jeweils durch Kommata getrennt und in Klammern eingefaßt sein müssen. Und jetzt kommt der springende Punkt. Bei der Definition des Makros beziehen sich die Übergabeparameter auf die im Makro definierten Label und Variablen. Bei einem Aufruf beziehen sich diese Übergabeparameter aber auf die im Hauptprogramm definierten Label und Variablen.

Um ein Gefühl und ein Verständnis für die Übergabeparameter zu bekommen, soll kurz erklärt werden, wie der Assembler die Parameter behandelt. Angenommen, in einer Makrodefinition wird ein Parameter »par1« benutzt und im Makroaufruf selbst steht an seiner Stelle nicht par1 sondern »BELIEBIG«, dann setzt der Assembler diese beiden Ausdrücke gleich. Das heißt er führt folgende Operation aus:

par1 = BELIEBIG

Daraus folgt, daß beide Variablen nach dem Makroaufruf den gleichen Wert enthalten. Als Beispiel möchte ich die 16-Bit-Addition hernehmen. Das Makro soll aber so gestaltet werden, daß kein Arbeitsregister (A,X,Y) zerstört wird.

```
10   —.MA ADDW (ADR1,ADR2,SUMME)
20   —      PHA
30   —      LDA     ADR1
40   —      CLC
50   —      ADC     ADR2
60   —      STA     SUMME
70   —      LDA     ADR1+1
80   —      ADC     ADR2+1
90   —      STA     SUMME+1
100  —      PLA
110  —.RT
```

In diesem Makro wird also zum Inhalt einer Adresse2 und Adresse2 +1 der Inhalt einer Adresse1 und Adresse1+1 addiert und das Ergebnis in der Adresse SUMME und SUMME+1 gespeichert. Um das Makro aufzurufen, ist das Programm wie folgt zu ergänzen:

```
120 —.BA $9000
130 —.EQ SUMMAND1 = $FB
140 —.EQ SUMMAND2 = $FD
150 —.EQ ERGEBNIS  = 2
160 —     LDA    #$E5
170 —     LDX    #$25
180 —     STA    SUMMAND1
190 —     STX    SUMMAND1+1
200 —     LDA    #$A8
210 —     LDX    #$43
220 —     STA    SUMMAND2
230 —     STX    SUMMAND2+1
240 —     ...    ADDW (SUMMAND1,
                 SUMMAND2,
                 ERGEBNIS)
250 —     LDX    ERGEBNIS
260 —     LDA    ERGEBNIS+1
270 —     JMP    $BDCD ;AUSGABE
                 AUF BILDSCHIRM
```

Dieses kleine Programm addiert die beiden Zahlen $25E5 und $43A8 und gibt das Ergebnis im dezimalen Format auf dem Bildschirm aus, das in diesem Fall 27021 ist. Um das zu testen, sind die Zeilen 10 bis 270 abzutippen, nachdem Hypra-Ass geladen und gestartet wurde. Ist der Quelltext assembliert, müßte, wenn kein Fehler aufgetreten ist, mit SYS 9*4096 das Ergebnis »27021« auf dem Bildschirm erscheinen. Um die Wirkung und die Arbeitsweise von Makros zu verstehen, spielen Sie mit dem Quelltext ruhig ein wenig herum. Ändern Sie einfach die Label und Zahlen ab, kaputtmachen können Sie nichts.

Hypra-Ass enthält noch eine weitere, sehr angenehme Eigenschaft. Er kann nämlich bedingt assemblieren. Was das ist und was man damit machen kann, soll hier detalliert beschrieben werden. Denn gerade im Zusammenhang mit Makros spielt die bedingte Assemblierung eine wichtige Rolle. Mit ihrer Hilfe kann bei einem Makroaufruf ein weiterer Parameter übergeben werden, der bestimmt, ob gewisse Quelltextteile innerhalb des Makros ins Hauptprogramm assembliert werden sollen oder nicht. Dadurch läßt sich in Abhängigkeit des Übergabeparameters ein Makro definieren, das verschiedene Aufgaben erfüllt. Fangen wir zunächst mit einem kleinen Beispiel an, damit Sie die Wirkung der bedingten Assemblierung innerhalb eines Makros kennenlernen. Dazu soll wieder die obenstehende 16-Bit-Addition herhalten, in die wir eine bedingte ».IF«-Abfrage einbauen wollen. Mit dieser bedingten ».IF«-Abfrage soll nun bestimmt werden, ob der Akku bei einem Makroaufruf erhalten bleiben oder zerstört werden soll, beziehungsweise ob das erzeugte Maschinenprogramm den »PHA«- und den »PLA«-Befehl enthalten soll oder nicht. Dazu sind zunächst die Zeilen 10 und 240 abzuändern, denn wir benötigen ja noch einen weiteren Übergabeparameter. Nennen wir diesen Übergabeparameter »RETTEN«. Dann sind die beiden Zeilen wie folgt zu ändern:

10 -.MA ADW (ADR1,ADR2,SUMME, RETTEN)
240 - ... ADW (SUMMAND1, SUMMAND2,ERGEBNIS,x)

Das »x« ist zu ersetzen durch eine »1« für Akku retten oder für eine beliebige andere Zahl für Akku zerstören. Als nächstes ist noch die bedingte ».IF«-Abfrage in den Quelltext einzubauen. Bevor das geschieht noch einiges zur ».IF«-Abfrage selbst, die etwas anders funktioniert, als Sie es vom Basic her ge-

Fortsetzung auf Seite 173



Bitte achten Sie darauf, nur die in den Jahrgangsübersichten aufgeführten Ausgaben zu bestellen.

Schaffen Sie sich ein interessantes Nachschlagewerk und gleichzeitig ein wertvolles Archiv!

Greifen Sie jetzt zu, solange ältere Ausgaben noch lieferbar sind!

Alle noch lieferbaren Ausgaben finden Sie in den Jahrgangsübersichten. Prüfen Sie, welche Ausgaben Ihnen in Ihrer Sammlung fehlen und die Sie deshalb nachbestellen wollen. Tragen Sie die Nummer der Ausgabe und das Erscheinungsjahr (z.B. 12/84) in dem Bestellabschnitt auf der Rückseite der untenstehenden Zahlkarte ein und geben Sie an, wieviele Exemplare dieser Ausgabe Sie bestellen. Die ausgefüllte Zahlkarte einfach heraustrennen und Rechnungsbetrag beim nächsten Postamt einzahlen. Ihre Bestellung wird sofort nach Zahlungseingang zur Auslieferung gebracht.

1984 JAHRGANGSÜBERSICHT

			4
			8
9	10		12

1985 JAHRGANGSÜBERSICHT

1	2	3	4
5	6	7	8
9	10	11	12

Lernen Sie die 64'er Programm-Sonderhefte kennen! Folgende Ausgaben sind noch lieferbar:

SONDERHEFT ABENTEUERSPIELE



Auf mehr als 100 Seiten viele interessante Adventures, die Sie lange Zeit fesseln werden. Mit abgeschlossenem Kurs zur Programmierung eigener Abenteuerspiele und zahlreichen Lösungen professioneller Adventures. BESTELLCODE: Abenteuerspiele

SONDERHEFT SPIELE



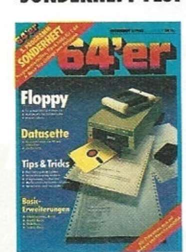
Heiße Listings für alle Spiele-Fans: Sportspiele, Spielegeneratoren, Abenteuerspiele, Taktikspiele, Geschicklichkeitsspiele und eine Marktübersicht bringen alles, was das Herz des Spiele-Fans höher schlagen läßt. BESTELLCODE: Spiele

SONDERHEFT GRAFIK&DRUCKER



Rund 28 Listings der Spitzenklasse. Top-Listings zur räumlichen Darstellung. Prüfsummenlistings, Drucker-Anwendung, Basic-Erweiterung, Hardcopy-Routinen, Zeichengenerator, Tips&Tricks. BESTELLCODE: Grafik&Drucker

SONDERHEFT FLOPPY/DATASETTE



So stellt man die Datasette ein. FMON 1541: Das Werkzeug für Floppy-Spezialisten. Disk-Basic 64: Fast 50 neue Befehle für komfortablen Floppy-Betrieb. Turbo Tape de Luxe: Datasette 10mal schneller als Floppy 1541. BESTELLCODE: Floppy

---

DM Pf für Postscheckkonto Nr. 14 199-803

Absender der Zahlkarte

Für Vermerke des Absenders

Postscheckkonto Nr. des Absenders

PSchA Postscheckkonto Nr. des Absenders Postscheckteilnehmer

Postscheckkonto Nr. des Absenders

Empfängerabschnitt

Zahlkarte/Postüberweisung

Die **stark umrandeten Felder** sind nur auszufüllen, wenn ein **Postscheckkontoinhaber** das Formblatt als **Postüberweisung** verwendet (Erläuterung s. Rücks.)

Einlieferungsschein/Lastschriftzettel

DM Pf

DM Pf (DM-Betrag in Buchstaben wiederholen)

DM Pf

für Postscheckkonto Nr.
14 199-803
Lieferanschrift und **Absender** der Zahlkarte

für Postscheckkonto Nr. **14 199-803** Postscheckamt **München**

für Markt&Technik Verlag Aktiengesellschaft
in **8013 Haar**

Postscheckkonto Nr. **14 199-803**
Postscheckamt **München**

für Markt&Technik Verlag Aktiengesellschaft
Hans-Pinsel-Str. 2
in 8013 Haar

PLZ Ort
Verwendungszweck
64'er
Leser-Service

Ausstellungsdatum Unterschrift

Postvermerk

Jetzt ist sie da: die Original

64'er -Sammelbox

Für alle Leser, die »64'er« regelmäßig kaufen, sammeln oder im Abonnement beziehen, gibt es jetzt ein interessantes Service-Angebot: Die 64'er-Sammelbox!

Mit dieser Sammelbox bringen Sie nicht nur Ordnung in Ihre wertvollen Hefte, sondern schaffen sich gleichzeitig ein interessantes und attraktives Nachschlagewerk.

Übrigens: Die Sammelbox ist nicht nur ein praktisches Aufbewahrungsmittel: Sie eignet sich auch hervorragend als Geschenk für Freunde und Bekannte zu vielen Anlässen.

Ein kompletter Jahrgang (12 Hefte) paßt in die praktische Sammel-Box! Am besten gleich bestellen!

Und so kommen Sie einfach und schnell zu Ihrer Sammelbox:

Vorbereitete Zahlkarte auf dieser Seite ausfüllen, Anzahl der gewünschten Sammelboxen, Sonderhefte oder älteren Ausgaben (s. Rückseite) angeben, Zahlkarte heraustrennen und Rechnungsbetrag beim nächsten Postamt einzahlen. Lieferung erfolgt sofort nach Zahlungseingang.

Wichtig: Es werden ausschließlich Bestellungen gegen Vorauszahlung mit Zahlkarte ausgeliefert. Ihre Bestellung wird sofort nach Zahlungseingang zur Auslieferung gebracht!

Bestellung Leser-Service

Für Mitteilungen an den Empfänger

Wichtig: Lieferanschrift (Rückseite) nicht vergessen!

Bestell-Nr.	Anzahl	x Einzelpreis	= Gesamtpreis
Sammelbox »64'er«		DM 14,–	
Sonderheft		DM 14,–	
Sonderheft		DM 14,–	
Ausg. ..../1984		DM 6,50	
Ausg. ..../1985		DM 6,50	
Zzgl. einm. Versandkostenpauschale (DM 3,–)			DM 3,–
Summe bitte auf Vorderseite übertragen.	Gesamtsumme:		DM

Hinweis für Postgirokontoinhaber:

Dieses Formblatt können Sie auch als Postüberweisung benutzen, wenn Sie die stark umrandeten Felder zusätzlich ausfüllen. Die Wiederholung des Betrages in Buchstaben ist dann nicht erforderlich. Ihren Absender (mit Postleitzahl) brauchen Sie nur auf dem linken Abschnitt anzugeben.

1. Abkürzung für den Namen Ihres Postgiroamts (PGiroA) siehe unten
2. Im Feld »Postgiroteilnehmer« genügt Ihre Namensangabe
3. Die Unterschrift muß mit der beim Postgiroamt hinterlegten Unterschriftsprobe übereinstimmen
4. Bei Einsendung an das Postgiroamt bitte den Lastschriftzettel nach hinten umschlagen

Abkürzungen für die Ortsnamen der PGiroA:

Bln W = Berlin West, Dtmd = Dortmund, Esn = Essen, Ffm = Frankfurt am Main, Hmb = Hamburg, Han = Hannover, Krlh = Karlsruhe, Kln = Köln, Lshfn = Ludwigshafen am Rhein, Mchn = München, Nbg = Nürnberg, Sbr = Saarbrücken, Stgt = Stuttgart

Feld für postdienstliche Zwecke

Bedienen Sie sich der Vorteile eines eigenen Postgirokontos

Auskunft hierüber erteilt jedes Postamt

Einlieferungsschein/Lastschriftzettel

(nicht zu Mitteilungen an den Empfänger benutzen)

Gebühr für die Zahlkarte (wird bei der Einlieferung bar erhoben)

bis 10 DM — 90 Pf

über 10 DM (unbeschränkt) — 1,50 DM

Bei Verwendung als Postüberweisung gebührenfrei



Fortsetzung von Seite 170

wohnt sind. Trifft das Basic-Programm auf eine Zeile, in der eine »IF«-Abfrage steht, wird zunächst überprüft, ob diese »IF«-Abfrage wahr oder falsch ist. Ist sie falsch, wird direkt zur nächsten Zeile verzweigt. Das ist bei einem Maschinenprogramm beziehungsweise dem dazugehörigen Quelltext nicht möglich, denn es darf ja pro Zeile immer nur ein Maschinenbefehl stehen. Deshalb muß bei der bedingten ».IF«-Abfrage noch ein weiterer Pseudo-Opcode vorhanden sein, der dem Assembler das Ende dieser ».IF«-Abfrage mitteilt. Dieses ist der ».EI«-Pseudo für »ENDIF«, der eine ».IF«-Abfrage abschließt. Bei Hypra-Ass existiert noch ein dritter Pseudo im Zusammenhang mit ».IF«-Abfragen der ».EL«-Pseudo für »ELSE«. Die Struktur innerhalb eines Programms sieht dann so aus:

```
.IF A !=! 1
.
.EL
.
.EI
```

Es handelt sich hier um eine »entweder oder«-Assemblierung. Entweder wird der Quelltextteil zwischen ».IF« und ».EL« (A=1) oder zwischen ».EL« und ».EI« (A< >1); in das Maschinenprogramm assembliert.

Doch nun wieder zurück zum Beispiel. Haben Sie die Zeilen 10 und 240 geändert, kann die ».IF«-Abfrage einbaut werden. Dazu sind folgende Zeilen zu ergänzen:

```
15 -.IF RETTEN !=! 1
25 -.EI
95 -.IF RETTEN !=! 1
105 -.EI
```

Bei RETTEN darf es sich nur um eine Variable handeln und nicht um eine Adresse oder Label.

Hypra-Ass enthält noch einige andere Pseudo-Opcodes zur bedingten Assemblierung:

.ON ausdruck,sprungziel: Ist der Ausdruck hinter .ON wahr, wird zur Zeile »sprungziel« verzweigt. Beispiel:

```
100 -.ON A !=! 1, 200
```

Ist A = 1, wird der Quelltextteil, der zwischen der Zeilennummer 100 und 199 steht ignoriert, also nicht assembliert.

.GO sprungziel: Unbedingter Sprung zur Zeile »sprungziel«. Beispiel:

```
100 -.GO 200
```

Der Quelltext, der zwischen der Zeilennummer 100 und 199 steht, wird auf keinen Fall assembliert. Daraus folgt natürlich, daß dieser bedingte Pseudo-Opcode nur in Verbindung mit einer ».IF«-Abfrage sinnvoll ist. Denn was nutzt ein Quelltext, der aufgrund eines unbedingten Sprungs generell ignoriert wird.

Ein kleines Beispiel soll den Sinn und Zweck dieses Pseudos verdeutlichen. Dazu wollen wir aus dem Quelltext heraus einen bestimmten Speicherbereich mit dem Maschinenbefehl »NOP« füllen.

```
10   —.BA $9000
20   —.EQ A = 0
30   —.IF A !<! 255
40   —                NOP
50   —.EQ A = A + 1
60   —.GO 30
70   —.EI
80   —                RTS
```

Beim Gebrauch von ».GO« und ».ON« ist anzumerken, daß im Falle eines RENUMBERs mit dem Editorbefehl »/N« die Sprungziele nicht angepaßt werden.

Was nun die bedingte Assemblierung leistet, soll an einem Makro, das einen »JSR«-Befehl und einen »JMP«-Befehl simuliert, der sämtliche Adressierungsarten beherrscht, gezeigt werden (Listing 1). Dabei wird ebenfalls ein Parameter übergeben, der die Adressierungsart bestimmt. Das Makro läßt sich vom Hauptprogramm mit dem Befehl »... JSR (adresse,adressierungsart)« aufrufen. Der »JMP«-Befehl wird genauso angewendet wie der »JSR«-Befehl. Adressierungsart bedeutet:

1 = X-indiziert; 2 = Y-indiziert; 3 = indiziert, indirekt; 4 = indirekt, indiziert

Was nun tatsächlich in das Maschinenprogramm assembliert wird, zeigt für den Fall, daß die Adressierungsart gleich 1 ist (also X-indiziert) Listing 2 und für den Fall, daß die Adressierungsart gleich 4 ist (also indirekt indiziert) Listing 3.

Sollten Sie noch irgendwelche Fragen zur Bedienung eines Assemblers haben, würden wir uns freuen, Ihnen bei der Lösung Ihrer Probleme zu helfen. Wir sind bestrebt, alle schriftlichen Anfragen zu beantworten. (ah)

```
30   -.MA JMP (ADRESSE,XY)
40   -.EQ HI = $FB
50   -.EQ LO = $FA
60   -.GO 131
70   -.MA JSR (ADRESSE,XY)
80   -.EQ HI = $FB
90   -.EQ LO = $FA
100  -              LDA #>(RUECKSP)
110  -              PHA
120  -              LDA #<(RUECKSP)
130  -              PHA
131  -.ON  (XY = 1)!O!(XY = 2) , 270
140  -.IF (XY = 1)
150  -              LDA ADRESSE+1,X
160  -              STA HI
170  -              LDA ADRESSE,X
180  -              STA LO
190  -.EI
200  -.IF (XY = 2)
210  -              LDA ADRESSE+1,Y
220  -              STA HI
230  -              LDA ADRESSE,Y
240  -              STA LO
250  -.EI
251  -.GO 530
270  -              INC ADRESSE
280  -              BNE LBL1
290  -              INC ADRESSE+1
310  -.IF (XY = 3)
320  -LBL1          LDA (ADRESSE,X)
330  -.EI
340  -.IF (XY = 4)
350  -LBL1          LDA (ADRESSE),Y
360  -.EI
380  -              STA HI
390  -              LDA ADRESSE
400  -              BNE LBL2
410  -              DEC ADRESSE+1
420  -LBL2          DEC ADRESSE
440  -.IF (XY = 3)
450  -              LDA (ADRESSE,X)
460  -.EI
470  -.IF (XY = 4)
480  -              LDA (ADRESSE),Y
490  -.EI
500  -.IF (XY=3)!O!(XY=4)
510  -              STA LO
520  -.EI
530  -              BNE LBL3
540  -              DEC HI
550  -LBL3          DEC LO
560  -              LDA HI
570  -              PHA
580  -              LDA LO
590  -              PHA
600  -RUECKSP       RTS
610  -.RT
```

Listing 1. Dieses Makro simuliert einen JSR-Befehl mit sämtlichen Adressierungsarten.

```
,9000  A9 90     LDA #90
,9002  48        PHA
,9003  A9 1C     LDA #1C
,9005  48        PHA
,9006  BD 01 C0  LDA C001,X
,9009  85 FB     STA   FB
,900B  BD 00 C0  LDA C000,X
,900E  85 FA     STA   FA
,9010  D0 02     BNE 9014
,9012  C6 FB     DEC   FB
,9014  C6 FA     DEC   FA
,9016  A5 FB     LDA   FB
,9018  48        PHA
,9019  A5 FA     LDA   FA
,901B  48        PHA
,901C  60        RTS
```

Listing 2. Eine »1« als Adressierungsart generiert dieses Maschinenprogramm. (Disassemblerausdruck SMON)

```
,9000  A9 90     LDA #90
,9002  48        PHA
,9003  A9 28     LDA #28
,9005  48        PHA
,9006  E6 02     INC   02
,9008  D0 02     BNE 900C
,900A  E6 03     INC   03
,900C  B1 02     LDA  (02),Y
,900E  85 FB     STA   FB
,9010  A5 02     LDA   02
,9012  D0 02     BNE 9016
,9014  C6 03     DEC   03
,9016  C6 02     DEC   02
,9018  B1 02     LDA  (02),Y
,901A  85 FA     STA   FA
,901C  D0 02     BNE 9020
,901E  C6 FB     DEC   FB
,9020  C6 FA     DEC   FA
,9022  A5 FB     LDA   FB
,9024  48        PHA
,9025  A5 FA     LDA   FA
,9027  48        PHA
,9028  60        RTS
```

Listing 3. Eine »4« als Adressierungsart generiert dieses Maschinenprogramm. (Disassemblerausdruck SMON)

Wir suchen die Anwendung des Monats

Anwendung des Monats, was ist das? Nun, Sie haben einen Commodore 64 oder einen C 128 und versuchen diesen irgendwie sinnvoll einzusetzen. Unter einer sinnvollen Anwendung versteht die 64'er-Redaktion alles, was beispielsweise Programme im häuslichen Bereich bewirken. Es kann sich dabei um die Berechnung der Benzinkosten für Ihren Wagen handeln, um ein eigenes Textverarbeitungsprogramm gehen, sich um die Verwaltung Ihrer Tiefkühltruhe drehen oder ein ausgeklügeltes Telefon- und Adreßregister sein.

Setzen Sie Ihren C 128/C 64 mehr oder weniger beruflich ein? Auch, oder vor allem, das ist eine sinnvolle Anwendung. Sie führen die Lohn- und Gehaltsabrechnung, Ihre Lagerverwaltung, die Bestellungen auf einem Commodore-Heimcomputer durch? So spezielle Anwendungen wie die Berechnung der Statik von selbstgezimmerten Regalen, von Klimadiagrammen oder Vokabellernprogrammen für den Schulunterricht oder die Zinsberechnung bei Krediten sind ebenfalls Themen, die mehr als konkurrenzfähig sind.

Uns ist die Anwendung des Monats **500 Mark** wert. Schreiben Sie uns, was Sie mit Ihrem Computer machen:
Markt&Technik Verlag Aktiengesellschaft, Redaktion 64'er,
Aktion: Anwendung des Monats, Hans-Pinsel-Str. 2, 8013 Haar bei München.

Einmal im Monat gibt es 2000 Mark für das Listing des Monats

Diese nicht einmalige Gelegenheit sollten Sie nutzen. Wie? Schicken Sie uns Ihr bestes selbst erstelltes Programm. Bei der Art des Programms sind wir nicht wählerisch.

Sie haben ein sehr gutes (Schieß-, Knobel-, Denk-, Action-, Abenteuer-)Spiel geschrieben: einschicken!

Sie verfügen über ein komfortables Disketten-Kopier-(Sortier) Programm mit einigen außergewöhnlichen Leistungsmerkmalen: einschicken!

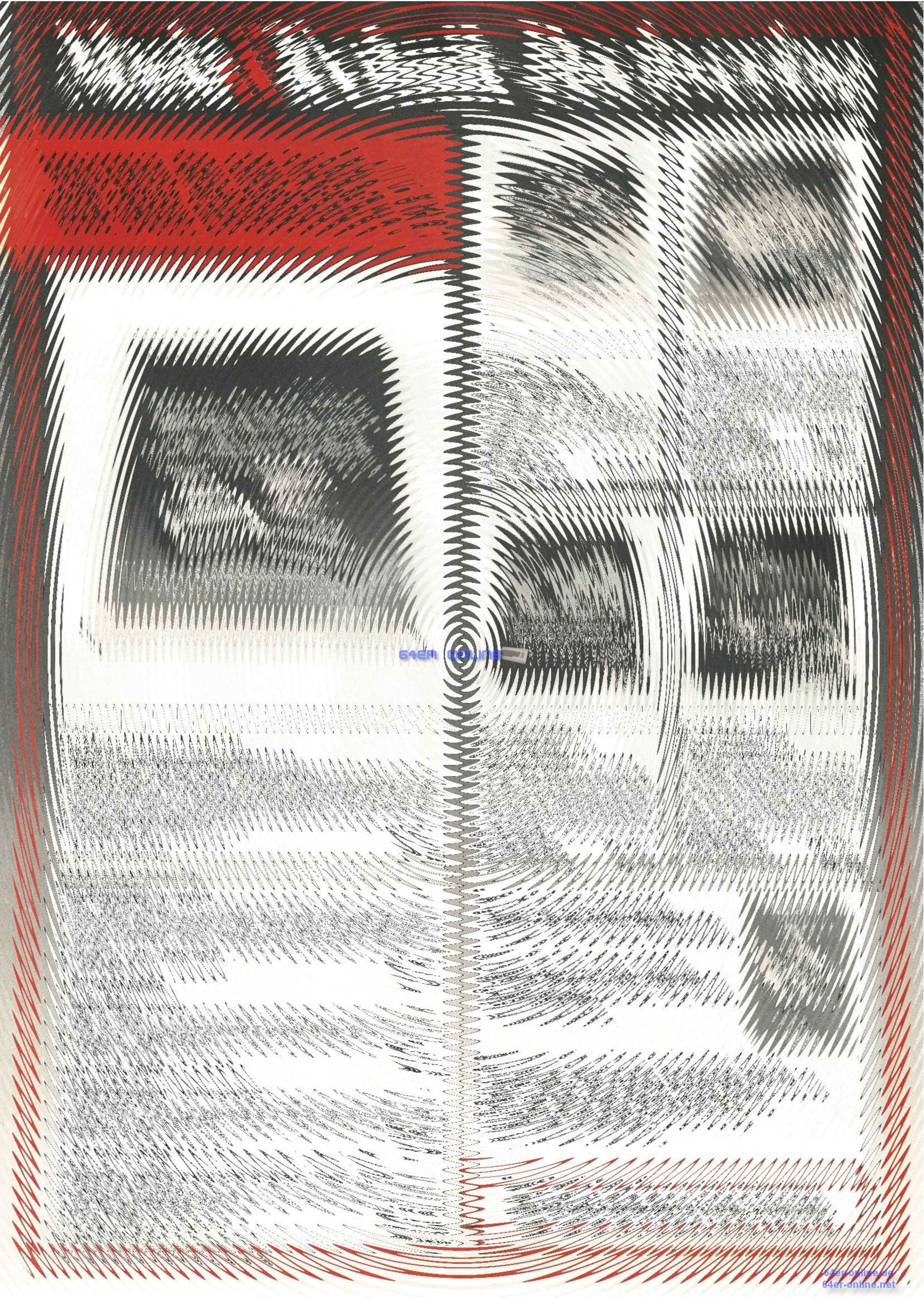
Sie haben das Basic um einige sinnvolle Befehle erweitert: einschicken!

Sie arbeiten mit einem selbsterstellten Textverarbeitungsprogramm, einer eigenen Tabellenkalkulation, einem semiprofessionellen Datenverwaltungsprogramm: einschicken!

Sie zeichnen und konstruieren mit einem selbsterstellten Programm in hochauflösender Grafik: einschicken!

Wir freuen uns über jeden Beitrag. Aus den besten Listings, die veröffentlicht werden, sucht die 64'er-Redaktion einmal im Monat das »Listing des Monats« aus. Alle Listings, die im 64'er abgedruckt sind, werden mit 100 bis 300 Mark honoriert. Die genaue Vorgehensweise beim Einsenden von Listings ist in dem Beitrag »Wie schicke ich meine Programme ein?« in verschiedenen Ausgaben beschrieben.

Schicken Sie Ihr Listing an: Markt&Technik Verlag Aktiengesellschaft, Redaktion 64'er,
Superchance: Listing des Monats, Hans-Pinsel-Straße 2, 8013 Haar bei München





Und wo lassen Sie reparieren?

In unserer Reparaturumfrage wollten wir wissen, was Sie im Falle eines Falles mit Ihrem defekten Gerät machen. Die zahlreiche Teilnahme zeigte, daß sich viele C 64-Besitzer über diese Frage Gedanken gemacht haben — hier sind einige Ergebnisse der Umfrage.

Computer gehören mit zu den zuverlässigsten elektronischen Geräten, die man überhaupt kaufen kann. Trotzdem kann es natürlich vorkommen, daß durch statische Entladung, Blitzschlag oder Fehlbedienung ein Computer nicht das tut, was er soll. Nicht ganz unschuldig sind auch manche Basteleien, die schon einigen Computern einen überdurchschnittlichen Alterungsprozeß beschert haben. In solchen Fällen ist es immer gut, wenn man einen schnellen und leistungsfähigen Reparatur-Service im Adreßbüchlein stehen hat, denn gerade bei einem Computer können falsche Reparaturen viel Schaden anrichten.

Ziel dieser Umfrage war es, festzustellen, wie der Reparatur-Service noch verbessert werden könnte. Am häufigsten wurde dabei der Wunsch geäußert, so kurz wie möglich auf den Computer (oder das Peripheriegerät) verzichten zu müssen.

Gleichzeitig sollte ein Reparaturservice natürlich preiswert sein. Viele Leser betrachten den Service auch in zunehmendem Maße als Kriterium für die Auswahl des Computers, sei es beim Erstkauf oder bei der Anschaffung eines leistungsfähigeren Systems. Als Ideallösung ist das früher übliche, aber bei steigender Verbreitung der Heim- und Personal Computer etwas in Vergessenheit geratene Austauschgerät anzusehen.

Wir sind sicher, daß Ihre Meinung nicht unberücksichtigt bleibt, denn ein guter Service ist der halbe Verkauf und gehört eigentlich zum Computer wie das Handbuch. Wir werden auf die entsprechenden Stellen einwirken, um eine Verbesserung der Reparatursituation in Deutschland zu erwirken, damit Sie in Zukunft nicht länger als unbedingt notwendig auf Ihr Gerät zu warten brauchen. Möglicherweise schaffen wir es sogar, den guten Brauch des Austauschgeräts wieder ins Gespräch zu bringen, denn nur so läßt sich Computer-Ausfallzeit vermeiden.

Für Ihre rege Mitarbeit gab es auch einiges zu gewinnen. Der Hauptpreis ist ein nagelneuer C 128, die Preise 2 bis 11 sind je ein Anwenderprogramm von Commodore. Der Gewinner des C 128 ist Herr Wilfried Christen aus Dattenberg — herzlichen Glückwunsch!

Je ein Commodore Magic Desk haben gewonnen: Harald Pinnow, Torsten Stracke, Andreas Urban, Sören von Otte. Je ein Calc Result von Commodore geht an Jürg Brik, Farine Robin. Die Gewinner der beiden Datenmanager sind Manfred Müller und Thomas Trenkwaldor. Die Programme Text 64 mit Adress 64 gehen je einmal an Erwin Wahsenbruck und Werner Braun. Ihnen allen herzlichen Glückwunsch und viel Freude mit Ihren neuen Programmen.

Alle Gewinner werden schriftlich benachrichtigt. Wir danken der Firma Commodore Büromaschinen GmbH für die Preise. (aw)

Die Computer und die Programme sind vom Umtausch ausgeschlossen.

Fortsetzung von Seite 67

der ursprüngliche Inhalt von Speicherzelle 1 zurückgelassen und ein Interrupt wieder erlaubt.

Die Addition der Nadelwertigkeiten der Punktzeilen findet in Zeile 3590 mit einem ADC-Befehl statt. Die Wertigkeiten der Nadeln stehen in der Tabelle NWERT am Schluß des Programms. Nachdem alle acht Werte im Akku summiert sind, positive Maskenvergleiche vorausgesetzt, wird der Akkuinhalt an den Drucker geschickt. Der Befehl JSR PRINT schickt das Punktmuster der Druckzeichenspalte an den Drucker. Sind die acht Punktreihen gedruckt, werden sämtliche Register wieder hergestellt und die Unterroutine AUSWERT wird beendet. Der RTS-Befehl führt dann wieder in die Spaltenschleife zu Zeile 2510.

Sind auf diese Weise 40 Zeichen gedruckt, wird in der Zeilenschleife das y-Register wieder auf »0« gesetzt, der Drucker auf eine neue Druckzeile mit Bitmusterdaten eingestellt und eine neue Druckzeile abgearbeitet.

Ist der ganze Bildschirm ausgedruckt (oder die STOP-Taste gedrückt), wird ab Zeile 2980 das Programm beendet. Im einzelnen heißt das, daß der Druckerkanal geschlossen, die Zeropage zurückgeschrieben und die normale Interruptroutine bearbeitet wird.

Danach läuft Ihr Programm weiter als wenn nichts geschehen wäre. (hm)

```
 programm : hardcopy.obj  c000 c12f
------------------------------------

c000 : 78 a9 0d a0 c0 8d 14 03   73
c008 : 8c 15 03 58 60 a5 c5 c9   c9
c010 : 04 d0 07 a9 00 85 c5 20   f7
c018 : 1d c0 4c 31 ea a2 ff b5   fe
c020 : 00 9d 2e c1 ca d0 f8 a9   1d
c028 : 01 85 cc a9 7e a2 04 a0   a2
c030 : 01 20 ba ff a9 00 20 bd   86
c038 : ff 20 c0 ff a2 7e 20 c9   aa
c040 : ff a9 1b 20 d2 ff a9 33   19
c048 : 20 d2 ff a9 18 20 d2 ff   d4
c050 : a9 00 ac 88 02 85 15 84   df
c058 : 16 a2 19 20 e1 ff f0 3f   6a
c060 : a0 27 b1 15 c9 20 d0 0b   9a
c068 : 88 10 f7 a9 0d 20 d2 ff   48
c070 : 4c 91 c0 20 09 c1 a0 07   e8
c078 : b9 1e c1 20 d2 ff 88 d0   a6
c080 : f7 a0 00 a9 00 b1 15 85   ea
c088 : d6 20 b7 c0 c8 c0 28 d0   49
c090 : f2 a9 28 18 65 15 85 15   a3
c098 : 90 02 e6 16 ca d0 bc a9   1f
c0a0 : 0d 20 d2 ff a9 7e 20 c3   08
c0a8 : ff 20 cc ff a2 ff bd 2e   68
c0b0 : c1 95 00 ca d0 f8 60 48   7c
c0b8 : 8a 48 98 48 a5 d6 85 f8   af
c0c0 : a9 00 85 f9 06 f8 26 f9   bf
c0c8 : 06 f8 26 f9 06 f8 26 f9   c8
c0d0 : a5 f9 18 65 fa 85 f9 a6   36
c0d8 : 01 a9 80 85 9d a9 00 48   36
c0e0 : a0 07 78 a9 01 29 fb 85   ab
c0e8 : 01 b1 f8 25 9d 86 01 58   67
c0f0 : f0 06 68 18 79 26 c1 48   61
c0f8 : 88 10 e7 68 20 d2 ff 46   b4
c100 : 9d 90 da 68 a8 68 aa 68   f2
c108 : 60 a9 00 85 f9 ad 18 d0   fd
c110 : 29 02 d0 05 a9 d0 85 fa   3c
c118 : 2c a9 d8 85 fa 60 00 01   b4
c120 : 40 04 2a 1b 0d 18 80 40   64
c128 : 20 10 08 04 02 01 00 07   [illegible]
```

MSE-Listing der Hardcopy-Routine

Inserentenverzeichnis

Impressum

Herausgeber: Carl-Franz von Quadt, Otmar Weber

Chefredakteur: Michael Scharfenberger (sc)
Leitender Redakteur: Albert Absmeier (aa)
Redakteure: ah = Achim Hübner, bs = Boris Schneider, aw = Arnd Wängler, ev = Volker Everts, gk = Georg Klinge, hm = Harald Meyer, og = Markus Ohnesorg, tr = Thomas Röder,
Redaktionsassistenz: Yvonne Runge (202)
Fotografie: Janos Feitser/Jens Jancke, Titelfoto: Jens Jancke

Layout: Leo Eder (Ltg.), Sigrid Kowalewski (Cheflayouterin), Dagmar Berninger, Willi Gründl

Auslandsrepräsentation:
Schweiz: Markt&Technik Vertriebs AG, Kollerstr. 3, CH-6300 Zug, Tel. 042-223155/56, Telex: 862329 mut ch
USA: M & T Publishing, 2464 Embarcadero Way, Palo Alto, CA 94303; Tel. (415) 424-0600; Telex 752351

Manuskripteinsendungen: Manuskripte und Programmlistings werden gerne von der Redaktion angenommen. Sie müssen frei sein von Rechten Dritter. Sollten sie auch an anderer Stelle zur Veröffentlichung oder gewerblichen Nutzung angeboten werden, so muß dies angegeben werden. Mit der Einsendung von Manuskripten und Listings gibt der Verfasser die Zustimmung zum Abdruck in von der Markt & Technik Verlags AG herausgegebenen Publikationen und zur Vervielfältigung der Programmlistings auf Datenträger. Mit der Einsendung von Bauanleitungen gibt der Einsender die Zustimmung zum Abdruck in von Markt&Technik Verlag AG verlegten Publikationen und dazu, daß Markt&Technik Verlag Geräte und Bauteile nach der Bauanleitung herstellen läßt und vertreibt oder durch Dritte vertreiben läßt. Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte und Listings wird keine Haftung übernommen.

Herstellung: Klaus Buck (180)

Anzeigenverkaufsleitung: Ralph Peter Rauchfuss (126)

Anzeigenverkauf: Brigitta Fiebig (282)

Anzeigenverwaltung und Disposition: Michaela Hörl (171), Liane Huber (168)

Anzeigenformate: 1/1-Seite ist 266 Millimeter hoch und 185 Millimeter breit (3 Spalten à 58 mm oder 4 Spalten à 43 Millimeter). Vollformat 297x210 Millimeter. Beilagen und Beihefter siehe Anzeigenpreisliste.

Anzeigenpreise: Es gilt die Anzeigenpreisliste Nr. 2 vom 1. Januar 1985.
Anzeigengrundpreise: 1/1 Seite sw: DM 8500,-. Farbzuschlag: erste und zweite Zusatzfarbe aus Europaskala je DM 1400,-. Vierfarbzuschlag DM 3800,-. Plazierung innerhalb der redaktionellen Beiträge: Mindestgröße 1/2-Seite
Anzeigen im Computer-Markt: Die ermäßigten Preise im Computer-Markt gelten nur innerhalb des geschlossenen Anzeigenteils, der ohne redaktionelle Beiträge ist. 1/1-Seite sw: DM 6400,-. Farbzuschlag: erste und zweite Zusatzfarbe aus Europaskala je DM 1000,-. Vierfarbzuschlag DM 3000,-. **Anzeigen in der Fundgrube: Private Kleinanzeigen** mit maximal 5 Zeilen Text DM 5,- je Anzeige.
Gewerbliche Kleinanzeigen: DM 11,- je Zeile Text.
Auf alle Anzeigenpreise wird die gesetzliche MwSt. jeweils zugerechnet.

Marketingleiter Vertrieb: Hans Hörl (114)

Vertriebsleitung: Helmut Grünfeldt (189)

Vertrieb Handelsauflage: Inland (Groß-, Einzel- und Bahnhofsbuchhandel) sowie Österreich und Schweiz: Pegasus Buch- und Zeitschriften-Vertriebsgesellschaft mbH, Hauptstätterstraße 96, 7000 Stuttgart 1, Telefon (0711) 6483-0

Erscheinungsweise: 64'er, Magazin für Computerfans, erscheint monatlich, Mitte des Vormonats.

Bezugsmöglichkeiten: Leser-Service: Telefon 089/4613-119. Bestellungen nimmt der Verlag oder jede Buchhandlung entgegen. Das Abonnement verlängert sich zu den dann jeweils gültigen Bedingungen um ein Jahr, wenn es nicht zwei Monate vor Ablauf schriftlich gekündigt wird.

Bezugspreise: Das Einzelheft kostet DM 6,50. Der Abonnementspreis beträgt im Inland DM 78,- pro Jahr für 12 Ausgaben. Darin enthalten sind die gesetzliche Mehrwertsteuer und die Zustellgebühren. Der Abonnementspreis erhöht sich um DM 18,- für die Zustellung im Ausland (Schweiz auf Anfrage), für die Luftpostzustellung in Ländergruppe 1 (z.B. USA) um DM 38,-, in Ländergruppe 2 (z.B. Hongkong) um DM 58,-, in Ländergruppe 3 (z.B. Australien) um DM 68,-.

Druck: E. Schwend GmbH, Schmollerstr. 31, 7170 Schwäbisch Hall

Verantwortlich: Für redaktionellen Teil: Michael Scharfenberger. Für Anzeigen: Brigitta Fiebig.

Redaktions-Direktor: Michael M. Pauly

Vorstand: Carl-Franz von Quadt, Otmar Weber

Anschrift für Verlag, Redaktion, Vertrieb, Anzeigenverwaltung und alle Verantwortlichen:
Markt&Technik Verlag Aktiengesellschaft, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon 089/4613-0, Telex 522052

Telefon-Durchwahl im Verlag:

Wählen Sie direkt: Per Durchwahl erreichen Sie alle Abteilungen direkt. Sie wählen 089-4613 und dann die Nummer, die in Klammern hinter dem jeweiligen Namen angegeben ist.

Mitglied der Informationsgemeinschaft zur Feststellung der Verbreitung von Werbeträgern e.V. (IVW), Bad Godesberg.



Die neueste Software für den C 128

Wir haben die erste speziell für den C 128-Modus entwickelte professionelle Software für sie getestet. Mit Protext und Superscript liegen zwei Textverarbeitungsprogramme der Sonderklasse vor, während es sich bei Superbase um eine klassische Dateiverwaltung handelt. Allen drei Programmen ist gemeinsam, daß sie den C 128 bis zur Grenze ausnutzen.

Preisdruck

Der Citizen 120 D ist ein kleiner Drucker, der einer von den ganz großen werden kann. Er bietet ähnliche Leistungen, wie der in dieser Ausgabe getestete MSP 10, kostet aber nur einen Bruchteil davon. Von einer Premiere kann man auch beim Fujitsu DX 2100 sprechen, denn wir erhielten eines der wenigen Exemplare der Vorproduktion zum Test.

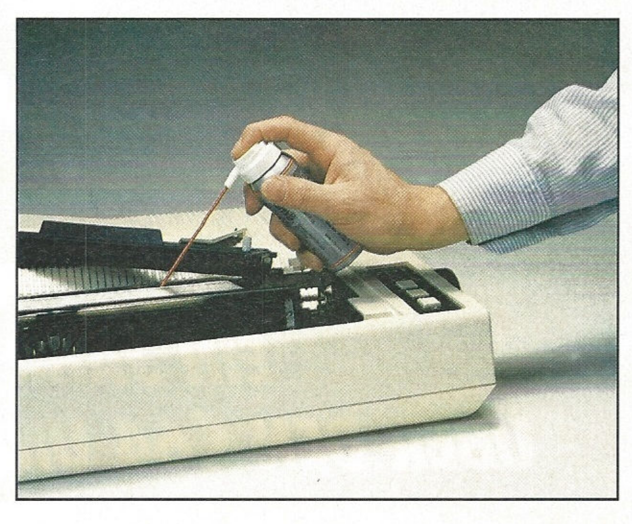
Neue Textadventures

In unserem Spieleteil stellen wir vier neue Textadventures vor, darunter das deutsche »Mord an Bord«, ein Adventure nach der Fernsehserie »Raumschiff Enterprise«, das neue Infocom-Spiel »Whishbringer« und die Adventure-Sensation aus den USA: »Mindwheel« von Synapse/Broderbund, ein Adventure mit geradezu unglaublichem Wortschatz und einer fantastischen Story.

Die CIA durchschaut

Die 6526-Portbausteine, die CIAs, des C 64 sind weithin bekannt. Viele C 64-Fans haben sie in schlechter Erinnerung. Denn diese beiden Bausteine im C 64 sind sehr empfindlich. Häufig verabschieden sie sich, wenn beim Betrieb des C 64 Joystick- oder Bus-Stecker gezogen oder eingesteckt werden.

In einem Artikel zeigen wir Ihnen, was alles in diesen Bausteinen steckt, und was man damit machen kann. Neben einigen Grundlagen geben wir Ihnen eine Registerübersicht der CIAs.

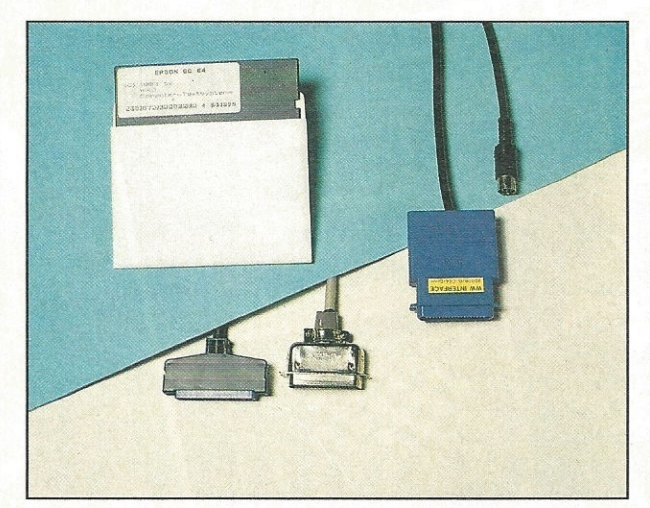


Drucker-Pflege

Ein Drucker ist ein mechanisches Präzisionsgerät, das eine regelmäßige Pflege erfordert. Sie erfahren, welche Teile gewartet werden müssen und welche nicht. Was ist zu tun, wenn das Druckbild sich, trotz eines frischen Farbbandes, verschlechtert? Solche und andere Fragen werden wir beantworten.

Textsysteme

Die Textverarbeitung auf dem C 64 ist eine Anwendung, bei der es, wie in kaum einem anderen Bereich, darauf ankommt, daß alle Teile des Computersystems bestens aufeinander abgestimmt sind. Sparen Sie sich viele Enttäuschungen und lesen Sie, welche Leistung Sie von den verschiedenen Textsystemen erwarten dürfen.



Guter Anschluß

Versteht Ihr Drucker Ihren Computer nicht? Erhalten Sie nicht den Ausdruck, den Sie erwartet haben? Wir räumen auf mit den Unsicherheiten beim Druckeranschluß. In einer übersichtlichen Darstellung erfahren Sie, welches Interface für Ihre Anwendung das am besten geeignete ist und welche Einschränkungen es gibt.

Tips zur Textverarbeitung

Sie werden feststellen, was in Ihrem Textverarbeitungsprogramm alles steckt. Wir geben nämlich Tips zu den bekanntesten Textverarbeitungsprogrammen. Auch bei der Auswahl des für Sie am besten geeigneten Programms helfen wir Ihnen, indem wir zeigen, welche Merkmale zu einer guten Textverarbeitung gehören und auf welche man eigentlich verzichten kann.

Software-Test: Newsroom

Gefällt Ihnen die 64'er nicht mehr? Wollen Sie Ihre eigene Zeitung machen? Dann sollten Sie wenigstens noch die nächste Ausgabe kaufen und unseren Software-Test über das Programm »Newsroom« lesen. Denn mit »Newsroom« können Sie kinderleicht Ihre eigene Zeitung produzieren. Benötigt wird nur ein C 64, ein Diskettenlaufwerk und ein Matrixdrucker.


DIE NETTESTE ART,
NETTE FREUNDE,
BEKANNTE
und VERWANDTE
EIN JAHR LANG AKTUELL
ÜBER DEN C64 UND SEIN UMFELD
ZU INFORMIEREN:
DAS »64'er«
GESCHENK-ABONNEMENT
Sie verschenken damit zwölfmal jährlich wichtige und aktuelle Informationen rund um den Commodore 64.
Das Geschenk-Abonnement von »64'er« bestellen Sie ganz einfach mit der nebenstehenden Bestellkarte.
Auf Wunsch überreichen wir dem Empfänger auch eine attraktive Geschenkurkunde.
Markt&Technik
64'er
DEZEMBER 1985
DM 6,50
12/85
DAS MAGAZIN FÜR COMPUTER-FANS
Monitore
Spiele
Top Ten '85
Software-Hilfen
Tips zu Vizawrite
Geschenk-Urkunde
64er-online.de
64er-online.net



